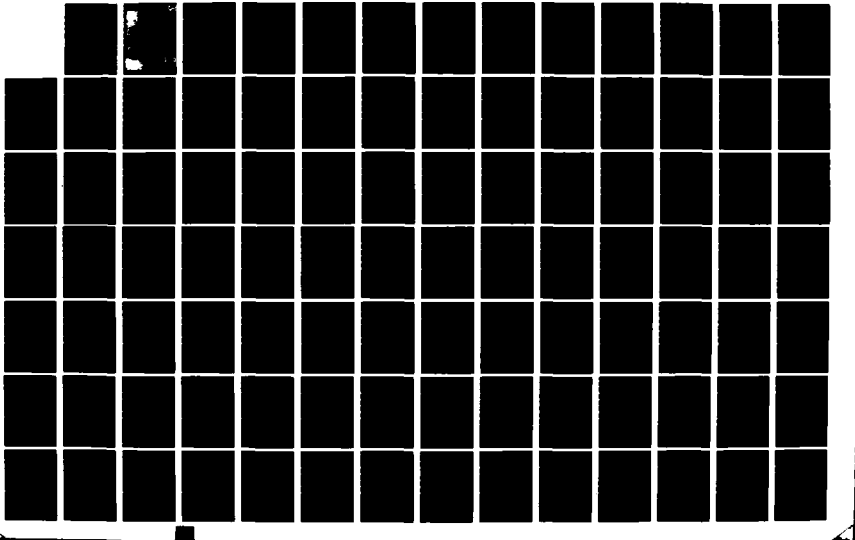
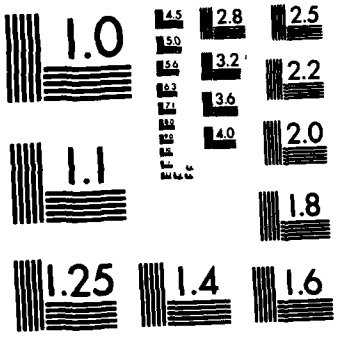
AD-A124 866 NUMERICAL IMPLEMENTATION OF THE COHESIVE SOIL BOUNDING SURFACE PLASTICITY..(U) CALIFORNIA UNIV DAVIS DEPT OF CIVIL ENGINEERING L R HERRMANN ET AL. FEB 83 1/2

UNCLASSIFIED NCEL-CR-83.010 N62474-82-C-8276 F/G 8/13 NL





CR 83.010

NAVAL CIVIL ENGINEERING LABORATORY
Port Hueneme, California

Sponsored by
NAVAL FACILITIES ENGINEERING COMMAND
Alexandria, Virginia 22332

NUMERICAL IMPLEMENTATION OF THE COHESIVE SOIL BOUNDING SURFACE PLASTICITY MODEL (VOLUME I)

February 1983

An Investigation Conducted by
DEPARTMENT OF CIVIL ENGINEERING
University of California
Davis, California

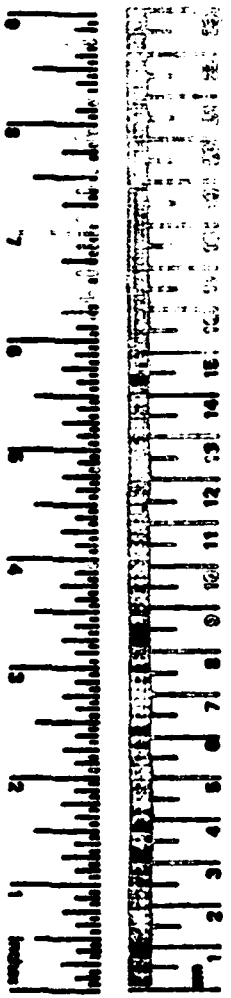
N62474-82-C-8276

Approved for public release; distribution unlimited

Approximate Conversions to Metric Measures

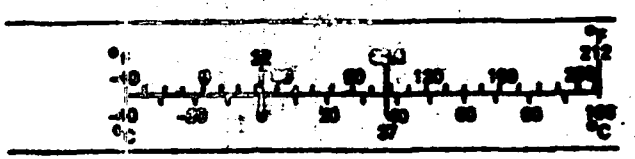
Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
in	inches	*2.5	centimeters	cm
ft	feet	30	centimeters	cm
yd	yards	0.9	meters	m
mi	miles	1.6	kilometers	km
		AREA		
in^2	square inches	6.5	square centimeters	cm^2
ft^2	square feet	0.09	square meters	m^2
yd^2	square yards	0.8	square meters	m^2
mi^2	square miles	2.6	square kilometers	km^2
	acres	0.4	hectares	ha
		MASS (weight)		
oz	ounces	28	grams	g
lb	pounds	0.45	kilograms	kg
	short tons (2,000 lb)	0.9	tonnes	t
		VOLUME		
tsp	teaspoons	5	milliliters	ml
Tbsp	tablespoons	15	milliliters	ml
fl oz	fluid ounces	30	milliliters	ml
c	cups	0.24	liters	l
pt	pints	0.47	liters	l
qt	quarts	0.95	liters	l
gal	gallons	3.8	liters	l
ft^3	cubic feet	0.03	cubic meters	m^3
yd^3	cubic yards	0.76	cubic meters	m^3
		TEMPERATURE (exact)		
°F	Fahrenheit temperature	5/9 (after subtracting 32)	Celsius temperature	°C

*1 in = 2.54 (exactly). For other exact conversions and more detailed tables, see NBS Misc. Publ. 286, Units of Weights and Measures, Price $2.25, SD Catalog No. C13.10:286.



Approximate Conversions from Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
mm	millimeters	0.04	inches	in
cm	centimeters	0.4	inches	in
m	meters	3.3	feet	ft
m	meters	1.1	yards	yd
km	kilometers	0.6	miles	mi
		AREA		
cm^2	square centimeters	0.16	square inches	in^2
m^2	square meters	1.2	square yards	yd^2
km^2	square kilometers	0.4	square miles	mi^2
ha	hectares (10,000 m^2)	2.5	acres	
		MASS (weight)		
g	grams	0.035	ounces	oz
kg	kilograms	2.2	pounds	lb
t	tonnes (1,000 kg)	1.1	short tons	
		VOLUME		
ml	milliliters	0.03	fluid ounces	fl oz
l	liters	2.1	pints	pt
l	liters	1.06	quarts	qt
l	liters	0.26	gallons	gal
m^3	cubic meters	35	cubic feet	ft^3
m^3	cubic meters	1.3	cubic yards	yd^3
		TEMPERATURE (exact)		
°C	Celsius temperature	9/5 (then add 32)	Fahrenheit temperature	°F



Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER CR 83.010	2. GOVT ACCESSION NO. AD-A124866	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Numerical Implementation of the Cohesive Soil Bounding Surface Plasticity Model (Volume I)		5. TYPE OF REPORT & PERIOD COVERED Not Final Apr 1982 - Nov 1982 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Leonard R. Herrmann, Prof.; Jay S. DeNatale, Asst. Prof.; Yannis F. Dafalias, Assoc. Prof.; Wern Ping Chen, Graduate Student		8. CONTRACT OR GRANT NUMBER(s) N62474-82-C-8276
9. PERFORMING ORGANIZATION NAME AND ADDRESS University of California Davis, CA		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS YF023.03.01.002
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Civil Engineering Laboratory Port Hueneme, CA 93043		12. REPORT DATE February 1983 13. NUMBER OF PAGES 114
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Naval Facilities Engineering Command 200 Stovall Street Alexandria, VA 22332		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Constitutive law, cohesive soil, finite element, numerical implementation

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The results of a study of various numerical means for implementing the bounding surface plasticity model for cohesive soils is presented. A comparison is made of the computational efficiency, robustness, and ease of implementation of the completing methods. The comparisons are made on the basis of solutions to three representative geotechnical engineering problems.

Accession For	
NTIS GRA&I	☒
DTIC TAB	☐
Unannounced	☐
Justification	
By	
Distribution/	
Availability Codes	
Dist	Avail and/or Special
A	

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

A very brief description of a computer-aided automated calibration procedure is given along with instructions for using the resulting computer code. An example calibration is reported for a particular cohesive soil.

TABLE OF CONTENTS

1. INTRODUCTION

A preliminary task of the project involved the continued evaluation of certain features of the model and the reporting of any additional theoretical developments.

Under other sponsorship, Professor Dafalias continued to study the predictive capabilities of the model for cyclic soil behavior. Preliminary results (Dafalias et al 1982) indicate the need for a mechanism to model the degradation of properties observed under long-term cyclic loading conditions. A possible means for modeling this phenomenon was developed and some preliminary testing was conducted. However, a complete validation of this modeling mechanism, and the determination of its usefulness, will require additional study.

Continued comparisons of model predictions and experimental results indicate the desirability of using a projection point other than the origin for the mapping rule. Accordingly, this feature has been included in the current version of the model. The interaction of this feature with other characteristics of the model is discussed in detail by DeNatale (1982).

Numerical evaluations of the model made during the course of the current study revealed some minor numerical problems associated with the first solution step away from a pure hydrostatic stress state for a normally consolidated soil. Professor Dafalias has proposed a slight modification of the model to remedy this problem. When this suggested change has been fully evaluated, and assuming that it does rectify the problem, it will be incorporated into the model.

The development of a computer aided, automated calibration scheme has been completed and used to perform a very extensive study of the importance and roles of the several material parameters which describe the model. A brief description of this study is given in the following section, and volume II of this report contains a user's manual and a listing for the calibration program. Before the end of the year, the

complete results of the study will be available in DeNatale's thesis (1982); a copy will be forwarded to NCEL at that time.

The main thrust of the current work has been the study of various schemes for numerically implementing the model in finite element analyses and the study of the numerical characteristics of the model. Particular attention was paid to the relative ease of implementation, and the economy and robustness of the competing schemes. In previous work (Herrmann et al 1981), the rate equations for the bounding surface model were cast in incremental form, and a subroutine was prepared to evaluate them. In Section 3, a very brief review of this step will be given, prior to the reporting on the main component of the research.

The final phase of the overall project will involve the verification of the predictive capabilities of finite element analyses which utilize the model. With this end in mind, a preliminary literature survey was conducted to determine the availability of laboratory and field experimental results. The findings of this search are given in Appendix A.

2. THE AUTOMATED CALIBRATION CODE

2.1 Purpose and Capabilities

In its most general form, the bounding surface model requires the determination of 19 separate constitutive parameters, including 2 initial state properties, 5 traditional material constants, whose values may be directly obtained from simple, easy to perform laboratory experiments, and 12 model constants, which must be indirectly established through a trial and error curve fitting process using the results of undrained triaxial tests. A general summary of the various properties is presented by Herrmann et al (1980), and a more detailed description of both the qualitative and quantitative influence of each parameter is provided by DeNatale (1982).

This breakdown of model constants is common to most, if not all, of the soil model formulations introduced in recent years. Determination of the directly

measureable or "fixed" parameters is straightforward and readily accomplished. Determination of the remaining "free" parameters, however, can make the calibration procedure prohibitively difficult. Rather than being measured directly from a particular portion of a specific laboratory test, these so-called "free" parameters must be determined by trial and error, with the objective being to obtain the best overall fit to a given experimental relation or set of observed responses. As a result, the overall accuracy and efficiency of the calibration process can be strongly dependent on both the subjectivity of the user as well as his expertise with the particular material model.

In formulations such as the bounding surface model, which employ a small number of material parameters whose roles in the constitutive formulation are each well defined, the manual calibration process becomes systematic and straightforward. However, reliance on user expertise is still high, since all manual curve fitting procedures, by their very nature, require both judgement (in deciding just what constitutes the "best" overall fit) and familarity (in deciding how much each parameter's value must be changed to improve a given prediction).

In order to simplify the model calibration process, a computer aided calibration procedure has been developed and tested. Since the calibration of a material model involves minimizing the error, or residual, between the observed and predicted soil response, the process can quite naturally be viewed as an optimization problem. Hence, the computer code employs a quasi-Newton optimization strategy to locate that set of parameter values which minimizes the discrepancy between the model predictions and the experimental observations included in the calibration data base. The code permits any number of tests, relations and/or individual observations to be included in the calibration data base. Different weights may be assigned to specific components of the data base if it is felt that certain tests, relations or observations are more reliable or representative than others, or if it is necessary to have the final model predictions fit some data more closely than others. Because this new computer aided

procedure greatly reduces the dependence of calibration success on user expertise, it significantly increases the accessibility and usefulness of sophisticated material models to the general engineering community. Although the code was developed specifically for use with the bounding surface plasticity model, it can be readily adapted to other constitutive formulations.

In addition to providing a means for automatically calibrating the bounding surface model, the code can also be used to generate a set of model predictions for homogeneous test conditions. The code may thus be used in those applications where the driving program EVAL and subroutine CLAY would formerly have been employed, Herrmann et al (1980). A comprehensive discussion of the new code, including a comparison of the effectiveness of the manual and automated calibration procedures, is presented by DeNatale (1982).

2.2 The Calibration Data Base

The ultimate goal of the calibration process is to identify that set of parameter values which enables the theoretical model to most closely simulate the observed material response. This goal is ordinarily accomplished by fitting the model to a representative set of experimentally observed stress-strain relations or "calibration data base". Ideally, this calibration data base should be complete and diverse enough that all important aspects of the material's response are included, and all necessary constitutive parameters may be uniquely established.

In its most general form, the bounding surface formulation is a fully three-dimensional stress-strain model. With a single set of parameter values the model may be applied to soil stress states at all overconsolidation ratios (OCR's), in either compression or extension, under both drained and undrained conditions, and for both monotonic and cyclic loading. Hence, to establish the optimal values of the necessary constitutive parameters, the calibration data base should ideally contain observations from the following seven standard laboratory tests:

1) an isotropic (or K_o) consolidation or drained compression test with both loading and unloading; and,

2-7) undrained triaxial compression and extension tests on specimens in the normally (OCR = 1), lightly ($1 < OCR \leq 2.5$) and heavily ($OCR \geq 4$) overconsolidated ranges.

The results of the consolidation test are required to establish the slopes of the isotropic consolidation and swell/recompression curves (in e - ln p' space) λ and κ. These two parameters are traditional soil properties and would normally be assigned values immediately, prior to using the automated calibration procedure. The results of the six undrained triaxial experiments are required to determine the 12 model constants cited in section 2.1, and would thus provide the data needed to direct the automated calibration procedure.

The triaxial test results should ideally be represented in terms of the observed q vs p', q vs ε_1 and u vs ε_1 relations. Naturally, if a less general form of the bounding surface model is acceptable, the number of constitutive parameters involved and the number of laboratory experiments required can be drastically reduced. For example, if the model is only to be applied to normally consolidated soils loaded in compression, the number of required constitutive parameters drops from 19 to 7, and only the isotropic consolidation and a single triaxial test are needed for model calibration.

Although the above data base is recommended, the bounding surface model could also be calibrated using other types of data. For example, drained rather than undrained tests could be employed. However, undrained tests are preferable, since good initial estimates of many of the model parameters can be made by examining the experimentally observed undrained stress paths.

There is also some evidence that the calibration data base need not necessarily include data from all three overconsolidation regions (see DeNatale 1982). That is, it may be sufficient to include only tests from the normal and heavy ranges, or perhaps

even the heavy range alone. The data which supports this possibility is not, however, conclusive, and therefore testing at all three regions is still recommended.

In addition, the experimental observations need not necessarily include all three relations q vs p', q vs ε_1 and u vs ε_1. Of the four undrained response parameters q, p', u and ε_1 only three are independent. In practice, p' is never actually measured, but rather is computed from the relation:

$$p' = \frac{3\ (\sigma_3 - u) + q}{3}$$

Hence, any two of the three relations cited above will completely define the soil response. The use of q vs p' or q vs ε_1 data only is insufficient, since each of these relations is insensitive to certain of the constitutive parameters. There is some evidence that the use of u vs ε_1 data alone may be sufficient (see DeNatale 1982). However, the use of all three response relations appears to increase the rapidity with which the optimization algorithm converges to the minimum. Presumably, the inclusion of redundant data reinforces the correct search direction. Since the cost of an automatic calibration run is only marginally affected by the numbers of response relations included in the calibration data base, it is recommended that all three of the relations cited above be used.

Finally, it may be possible to use other than triaxial tests to acquire the necessary experimental information. Although the conventional triaxial apparatus is the most common and versatile laboratory device, the simple shear apparatus could also, for example, be used. In general, the soils observed stress-strain characteristics will be to some extent dependent on the testing device which is used. Thus, in practial problems, the laboratory device used to generate the calibration data base should simulate, as closely as possible, the loading conditions for which the bounding surface predictions will eventually be generated.

2.3 Practical Considerations

The goal of the calibration procedure is to identify that set of parameter values which minimizes the error between the model predictions and the experimental observations included in the calibration data base. The automated optimization code thus seeks to locate the objective function's global minimum. Unfortunately, there is no guarantee that the algorithm will always succeed. The quasi-Newton strategy employed by the model calibration code, like most, if not all, practical optimization algorithms, is designed only to locate local minima in the vicinity of the initial estimates. Hence, the probability that the true global minimum will be found is directly related to the degree of unimodality exhibited by the objective function and the accuracy of the initial guess.

Preliminary research by DeNatale (1982) indicates that the use of the "absolute - Euclidean" measure of error leads to a more unimodal, and thus desireable, objective function. A procedure for acquiring improved starting estimates has also been developed by DeNatale (1982). Through actual testing with a number of different soils, this strategy has been found to produce starting estimates which enable the automated calibration code to consistently locate the global minimum. In practice, however, the only way to ensure that the global minimum has been found is to conduct the search from a variety of different starting points. The solution which yields the lowest value of the objective function may then be regarded as the global minimum.

A second practical consideration concerns the quality of the calibration data base. The user should ensure that the experimental observations included in the calibration data base are diverse enough to permit the optimal values of the required unknown model parameters to be uniquely defined. For example, if the code is used to identify those model parameters associated with heavily overconsolidated material response, the calibration data base must include observations made on heavily overconsolidated specimens. If the necessary experimental data is not included, the

program will continue to execute, but the final computed "optimal" values of the undefined parameters will be very close to the initial estimates. The major consequence of an inadequate or incomplete calibration data base is related to the cost of the analysis. Certain computational costs increase in proportion to n^2 (n = number of parameters to be determined), and a single gradient evaluation requires either n or 2n additional objective function evaluations (depending on whether forward or central differencing formulae are used). Thus, to minimize the cost of a model calibration run, the user should seek to identify only those parameters whose values can be defined, given the particular data base. A comprehensive discussion of the influence of each of the 19 model parameters is provided by DeNatale (1982), and may be referred to if uncertainly exists.

2.4 Example

To verify the viability of the new computer aided calibration procedure, the method was applied to a number of representative data bases, both artificial and real. The outcome of these studies is discussed by DeNatale (1982). Among the real data bases to which the automated process was applied are the experimental results on Kaolin reported by Jafroudi (1982).

The Bounding Surface model was calibrated on the basis of conventional undrained triaxial compression and extension tests on samples at overconsolidation ratios of OCR = 1, 2 and 6. With the necessary constitutive parameters thus having been fixed at their optimal values, predictions were then generated for a variety of additional undrained, drained, hollow cylinder and cyclic tests.

The results obtained with the automated procedure are compared to those of the manual solution, as reported by Jafroudi (1982) and Herrmann et al (1982), in Table 1 and Figures 1 through 6. As may be observed in Table 1, the optimal values of the model parameters as established through the automated and manual procedure are, as a group, distinctly different. For the given set of options specified in Table

1, the automated solution was better than the hand solution, in the sense of having a lower objective function value. The relative merits of the two solutions may perhaps most clearly be seen by comparing the associated calibration curves shown in Figures 1 through 6. On the whole, the automated curves appear to more closely match the experimental observations. A point by point comparison of the two solutions is essentially impossible, since in performing a manual calibration, different subjective weights are implicity assigned to the various components of the data base which cannot be precisely identified.

2.5 Cost

The automated calibration code has been written in FORTRAN and implemented or both an LSI-11/23 minicomputer as well as a VAX-11/780 super-minicomputer. The cost of a given analysis is controlled primarily by the number of distinct experimental tests included in the calibration data base and the number of constitutive parameters whose optimal values are being sought. A typical computer calibration, such as the 11-dimensional problem reported herein for the data of Jafroudi (1982), requires from 400-600 objective function evaluations, or about 60-90 minutes of VAX CPU time, at a cost of approximately $50.00-$75.00. This cost is relatively low when compared to the cost of the experimental program needed to establish the calibration data base, and in light of the resulting economy of finite element analyses based on the model. Calibration, of course, need only be done once for a given soil, regardless of the number and variety of subsequent finite element analyses which utilize the model.

3. NUMERICAL IMPLEMENTATION

3.1 Incrementalization Of Rate Equations

Using tensor notation, the basic rate equations for the bounding surface plasticity model can be written in the form (repeated indices are summed from 1 → 3 and free indices take on values of 1, 2 or 3.)

$$\dot{\sigma}_{ij} = D_{ijk\ell}\dot{\epsilon}_{k\ell} \tag{1}$$

The effective stress components are denoted by σ_{ij} and the strain components by ϵ_{ij}. In general, $D_{ijk\ell}$ is a function of both σ_{ij} and ϵ_{ij} and one or more internal variables q_i. The specific form of $D_{ijk\ell}$ for the bounding surface model for cohesive soils is given by Herrmann et al (1980).

For numerical analysis purposes, it is more convenient to express the relationship in matrix form; i.e.,

$$\{\dot{\sigma}\} = [D]\ \{\dot{\epsilon}\} \tag{2}$$

where ($[\ ]^T$ is the matrix transpose)

$$\{\sigma\}^T = (\sigma_x,\ \sigma_y,\ \sigma_z,\ \tau_{xy},\ \tau_{xz},\ \tau_{yz})$$

$$\{\epsilon\}^T = (\epsilon_x,\ \epsilon_y,\ \epsilon_z,\ \gamma_{xy},\ \gamma_{xz},\ \gamma_{yz}) \tag{3}$$

The tensor components of shear strain ϵ_{ij} are one-half of the engineering components γ_{ij}. The symmetric 6x6 [D] matrix is expressed in terms of the components of the 3x3x3x3 $D_{ijk\ell}$ tensor by using the following six sets of corresponding indices (1;1,1) (2;2,2), (3;3,3), (4;1,2), (5;1,3) and (6;2,3) where the first number is the row (column) number in the matrix and the second and third numbers are the first (last) two indices for the tensor.

To be able to use eq. (2) in an incremental solution procedure, it must be expressed in an incremental form. Consider the n^{th} step of an incremental analysis; i.e., the solution has been found at n-1, and it is desired to calculate the incremental change that will give the solution at n. Because of the nonlinear behavior, iteration is required to establish the incremental change. In the $k\text{-}1^{th}$ iteration of this process, the estimates of the stress and strain states at n are given by

$$\{\sigma\}_{n,k-1} = \{\sigma\}_{n-1} + \{\Delta\sigma\}_{n,k-1} \tag{4}$$

$$\{\epsilon\}_{n,k-1} = \{\epsilon\}_{n-1} + \{\Delta\epsilon\}_{n,k-1} \tag{5}$$

Even though rate independent behavior is being considered, it is convenient to think in terms of the time histories of the quantities involved. Integrating eq. (2) from time t_{n-1} to t_n, and using the trapezoidal formula to approximate the right hand side, gives

$$\{\Delta\sigma\}_{n,k} = [\bar{D}]_{n,k-1} \{\Delta\epsilon\} \tag{6}$$

$$[\bar{D}]_{n,k-1} = \frac{1}{2} \left[[D]_{n-1} + [D]_n \right]_{,k-1} \tag{7}$$

When eq. (6) is used in a finite element analysis it is tacitly assumed that all the stress and strain components at a particular point in the body, and from point to point, change proportionally from their values at n-1 to their values at n. Thus, in order that the true solution history be accurately modeled, as required for an inelastic material, the solution step size must be limited.

Eq. (6) is the desired incremental stress-strain equation for iteration k of increment n. Because $[D]_n$ is a function of the stress and strain states at n (due to the dependence of $D_{ijk\ell}$ on σ_{ij} at ϵ_{ij}), it is necessary to base its value on the estimates of the previous iteration (eqs. (4) and (5)). The resulting value, denoted by $[D]_{n,k-1}$, is used in eq. (7). The fact that $[D]_{n-1}$ is also a function of the stress and strain states at n is less obvious. This dependence arises because of the difference in material response for loading and unloading conditions and the fact that whether loading or unloading occurs during a given increment is influence by the values of $\{\sigma\}_n$ and $\{\epsilon\}_n$. That is, $[D]_{n-1}$ is the tangent stiffness at the beginning of the increment, and this stiffness differs for loading and unloading conditions, as determined by the values of $\{\Delta\sigma\}_n$ and $\{\Delta\epsilon\}_n$. The appropriate value of $[D]_{n-1}$ is written as $[D]_{n-1,k-1}$.

At the beginning of the iteration process, initial estimates are required. For the first iteration ot the first increment, they are usually taken to be zero. For the initial iterations of succeeding increments, they also can be started at zero; however,

it may be desirable to make use of information from the previous increment to obtain better starting values. The simplest procedure is to use as the initial estimate the final values found in the previous increment. This practice is based on the assumption of relatively uniform behavior from increment to increment. Difficulties can arise when the histories of the applied external agents acting on the structure cause a switch from loading to unloading in an unstable material response regime, and these external agents are loads not displacements. For example, consider the one-dimensional response shown in Figure 7. Consider the case when the state of the soil is at point "A" at the end of increment n-1. If during increment n, $\Delta\sigma_n$ is specified, two final states B and B' are possible. One corresponds to $\Delta\varepsilon$ (negative) and the other to $\Delta\varepsilon'$ (positive). Without any additional information, no choice can be made between B and B'. (It is easily seen that for specified stress increments in the stable region of behavior and for specified strain increments anywhere, no such problem exists.) The suggested solution to this impasse is to assume that the user would not attempt a stress controlled specification for "loading" conditions (path A-B') in an unstable region and, hence, if the stress increment is specified, unloading is the proper behavior (path A-B)[†]. For stress controlled conditions, the selection of the unloading path can be assured if the starting estimate of strain is of opposite sign to that calculated in the previous increment. Thus, the following strategy is recommended. When considering a series of increments for which the rates of the externally applied loads and displacements do not change sign[††], $\{\Delta\varepsilon\}_{n-1}$ and $\{\Delta\sigma\}_{n-1}$ are used as starting estimates for increment

---

[†] This argument requires that the arrival at A must have been preceeded by strain controlled steps.

[††] It is assumed that this condition is sufficient to prevent a general switch from loading to unloading within the soil mass.

n. However, as one such solution history segment is ended and a new one begins, the conditions necessary for the non-uniqueness problem may occur. Hence, for the first increment of each such series, it is suggested that the starting strain estimate be taken as some small negative multiple (e.g. -.01) of the value found in the previous increment (the stress increment would be used unchanged). The reduction in absolute magnitude is in deference to the greater stiffness encountered in unloading. Such an initial estimate will force the solution to select path A → B if the necessary conditions exist for the above mentioned non-uniqueness to occur. If non-uniqueness is not a problem, the only effect of this procedure is to slightly slow the convergence process.

It is important to note that, in general, the estimates of the stress and strain increments used in the calculation of $[\bar{D}]_{n,k-1}$ do not in fact satisfy eq. (6). The consequences of this inconsistency will be discussed later.

A FORTRAN subroutine CLAY for the calculation of the matrix $[\bar{D}]_{n,k-1}$ has been written and is documented by Herrmann, et al (1981).

3.2 Calculation Of Pore Water Pressure

The bounding surface plasticity theory is expressed in terms of effective stress, whereas most soil related problems involve the application and calculation of total stress. The total and effective stresses differ by the pore water pressure u. There are three possibilities concerning the development of pore water pressure in soil: ideal drained conditions (where the pore water pressure is identically zero), ideal undrained conditions (where the soil is completely saturated, and no flow of water occurs), and the more realistic situation where there is a global flow of water and/or the filling of voids. In many analyses ideal drained or undrained conditions are assumed, even though they may only be approximately true.

The total stress increment $\overset{\circ t}{\sigma}_{ij}$ is the sum of the effective stress increment and the pore water pressure increment:

$$\dot{\sigma}^{t}_{ij} = \dot{\sigma}_{ij} + \dot{u}\,\delta_{ij} \tag{8}$$

For drained conditions u=0 and $\sigma^{t}_{ij}=\sigma_{ij}$, and eq. (6) is the desired relationship between the total stress and strain increments.

For undrained conditions there are several possible ways of proceeding. The traditional approach has been to neglect the (slight) compressibility of the water and the soil particles, and thus assume incompressible material behavior. However, the finite element analysis of incompressible materials requires a special formulation (Herrmann 1965; Zienkiewicz 1977).

In order to avoid having to deal with separate formulations for drained and undrained conditions, it is convenient to express them in a common form. This can be accomplished if the slight compressibility of the soil particles and the pore water is recognized, Sangrey, et al (1969). (An alternative interpretation is to consider the undrained soil as incompressible, and to approximately specify the condition by means of a "penalty function", Zienkiewicz, et al 1981, where the associated "penalty number" corresponds to the bulk modulus Γ). Thus, the pore water pressure u is written in terms of the combined bulk modulus Γ of the soil particles and the pore water, and the resulting (very small) volume change ϵ_{kk} i.e., $u=\Gamma\,\epsilon_{kk}$. As $\Gamma\to\infty$ the soil becomes incompressible. Drained conditions are obtained when $\Gamma=0$. For undrained conditions the value of Γ is very large compared to the terms in $[\bar{D}]_{n,k-1}$. Thus, the soil behaves as a "nearly incompressible solid" (Herrmann 1965) and care must be exercised to avoid numerical round-off and excessive constraint problems. Two approaches are commonly used to achieve this goal. One method is to use the special formulation given by Herrmann (1965) for incompressible and nearly incompressible solids, while the other is to use "reduced" or "selective-reduced" integration (Zienkiewicz 1977) for the element stiffness matrix; the importance of selecting a proper element type, in order to achieve acceptable accuracy, is discussed by Nagtegaal, et al (1974) and Zienkiewicz, et al

(1981). In the latter case, the above expression is used to eliminate $\dot{u}$ from eq. (8); i.e.,

$$\dot{\sigma}^{t}_{ij} = \dot{\sigma}_{ij} + \Gamma \dot{\varepsilon}_{kk}\delta_{ij} \tag{9}$$

Integration over increment n gives:

$$\Delta\sigma^{t}_{ij_n} = \Delta\sigma_{ij_n} + \Gamma \Delta\varepsilon_{kk_n} \delta_{ij} \tag{10}$$

Eliminating $\Delta\sigma_{ij}$ using eq. (6) and returning to matrix notation yields:

$$\{\Delta\sigma^{t}\}_{n,k} = \bar{D}^{t}_{n,k-1}\{\Delta\varepsilon\}_{n,k} \tag{11}$$

where

$$[\bar{D}^{t}]_{n,k-1} = [\bar{D}]_{n,k-1} + [d] \tag{12}$$

All components of [d] are zero, except $d_{11}=d_{22}=d_{33}=\Gamma$.

Returning to the more realistic situation where water movement takes place, two cases can be distinguished. The first occurs when there is no global movement of water (either the time scale is too short for significant flow to occur, or the soil is stressed homogeneously, thus producing no pressure gradients) in a partially saturated soil. This condition can be modeled by using a variable Γ which is a function of the current saturation state.

When there is actual global flow of water, it is necessary to perform a coupled flow-stress analysis (Sandhu and Wilson 1969). The details of such an analysis are beyond the scope of this study. It should be noted that the bounding surface plasticity model is valid for such situations; however, the pore water pressure can not simply be calculated from the expression $\Gamma\varepsilon_{kk}$. Eq. (6) is, however, still valid for relating the increments of effective stress and strain.

3.3 General Discussion Of Finite Element Implementation Of Plasticity Models

The desired characteristics of any numerical scheme are ease of implementation, computational efficiency and robustness of behavior. It is on the bases of these three characteristics that the competing methods will be judged. There is of course a considerable body of literature available on the analysis of elasto-plastic bodies. Only those items which directly relate to this work will be discussed here; for a more general discussion the reader is referred to Owen and Hinton (1980) and Nagtegaal et al (1974).

In general, the response of an elasto-plastic body is highly nonlinear and path dependent. Thus, a general numerical analysis procedure for elasto-plasticity problems requires an incremental solution. Unless the increments are made excessively small, iteration must be conducted in each increment to account for the nonlinear behavior. The two most commonly employed classes of iterative methods are successive approximation (substitution) and Newton *like* procedures. The method of successive approximation can be cast in a variety of forms and thus is not a unique operation (see Isaacson and Keller 1966). The alternative forms range from extremely simple, but very slowly convergent, procedures, to more complicated methods. The Newton-Raphson method and numerous approximations to it constitute the class of "Newton like" methods. The many available solution methods obviously could not all be evaluated in this study. The selection of the methods that are compared in this study is discussed in the next section.

3.4 Selection Of Methods For Comparison

While the method of successive approximation is extremely easy to implement, it often suffers from poor convergence characteristics. For nonlinear elasticity problems, where the entire solution history can be accounted for in one step, there appears to be little question, unless the nonlinearities are very weak, that a Newton like method is to be preferred. However, for inelasticity problems where relatively

small steps are required to account for the path dependence of the solution, the choice is not so clear. The question is further clouded by the fact that, for a given solution step, and at a given point in an elasto-plastic body, the stiffness may be discontinuous due to the onset of yielding or to the progression from (local) loading to unloading conditions. The latter case is of concern in highly statically indeterminate situations where, at a given point in space and time, it is not known a priori whether or not the material will experience loading or unloading. This stiffness discontinuity may make convergence of Newton like method slower than for problems where the Jacobian is strictly continuous.

In order to keep the scope of the study within practical bounds, certain acceptability criteria are stated and used to limit the number of methods to be compared. Two of the key reasons for the computational efficiency of the finite element method, as applied to structural problems, are the symmetry and banded nature of the simultaneous equations. Hence, for this study, it is required that the solution methods preserve these characteristics.

It is on the basis of maintaining symmetry that Owen and Hinton (1980) rule out the general form of the Newton-Raphson method, and instead advocate the use of an approximate form; i.e. the "tangent stiffness" method.

Currently there is considerable interest in quasi-Newton methods for nonlinear structural problems (see Geradin et al 1981). The central goal of quasi-Newton methods is to avoid calculating the Jacobian every iteration. Instead, simple updating formulas are used to approximate the Jacobian (or its inverse) in terms of the previous approximation and simple vector quantities (the previous solution and current residual). The updates can be made directly to the Jacobian or to its inverse; in the first case, a set of simultaneous equations must be solved each iteration, while in the second, only a matrix multiplication is required. In the former case, because of the continued need of solving a set of simultaneous equations at each iteration, and because the cost

of using the updating formula is as great as the cost of calculating the approximate Jacobian for the other methods considered in this study, there would seem to be little or no advantage offered by the method (although in optimization and nonlinear elasticity problems, other important advantages make it a viable method). Thus, only those quasi-Newton methods which update the inverse would appear to be of interest in elasto-plasticity problems. Inverse updating methods, however, result in dealing with a full matrix, thus destroying the advantages of the banded characteristic of finite element equations. The ideal situation would be an updating formula which can be applied directly to the reduced (upper triangular form) Jacobian and which would not destroy its banded nature nor its original symmetry (a characteristic required for the efficient reduction, each iteration, of the load vector). The available quasi-Newton schemes of this type do not satisfy the ease of implementation requirement (see the discussions by Geradin et al 1980, 1981, Mathies and Strang 1979 and Schubert (1970). Thus, a simpler form is considered herein. Finally it should be noted that because of the lack of a natural objective function, the line search criterion of the quasi-Newton methods, as applied to optimization (see Fletcher 1980) and nonlinear elasticity, is lost for inelasticity problems. While alternative criteria have been proposed for elasto-plasticity problems, they appear to lack the simplicity and robustness of the minimization of an objective function.

The chief appeal of inverse quasi-Newton methods is the elimination of the necessity of reducing the Jacobian every iteration. This same objective can be achieved by using a modified iteration method which only occasionally updates the Jacobian. At other times the effects of the changes in the Jacobian are estimated and transferred to the right-hand side of the equations. This procedure is highly recommended by Owen and Hinton (1980) and is included in this evaluation.

Successive approximation methods are popular because of their simplicity of implementation. Thus one form of successive approximation is studied. In order to

demonstrate its relation to the Newton-Raphson method it is derived as an alternative to the tangent stiffness method.

In order to improve the rate (and also to enlarge the domain) of convergence of the iteration process, acceleration schemes are often used. These schemes usually employ some type of extrapolation in order to obtain a better solution estimate than given directly by the iteration process (see Isaacson and Keller 1966). The extrapolation can usually be epxressed in terms of an acceleration (iteration, relaxation) factor. The simplest method employs a constant factor selected by the user on the basis of past experience. Because, for inelasticity analyses, the optimum factor can vary widely from problem to problem, and even from increment to increment, some type of strategy for automatically selecting it is desirable. For one degree of freedom problems, the Aitken's ∇^2 method (see Isaacson and Keller 1966) is simple to apply and has proven to be quite effective. Two methods for adapting it to multi-degree of freedom problems are considered herein.

The final question that is addressed is how best to handle the inconsistency (previously mentioned) between the estimates $\{\Delta\sigma\}_{n,k-1}$ and $\{\Delta\varepsilon\}_{n,k-1}$ used to calculate $[\bar{D}]_{n,k-1}$ and eq. (6); three schemes are compared.

3.5 Theory

3.5.1 Solution Methods

In the following discussion it will be assumed that the reader is familiar with the standard steps involved in formulating a finite element analysis.

For a given solution increment eq. (6) is used in the formulation of element stiffness matrices, which are in turn combined to form the system stiffness matrix $[\bar{K}]_n$. The incremental load vector is denoted by $\{\Delta F\}_n$. Equilibrium leads to the following system of simultaneous equations for the incremental displacements $\{\Delta u\}_n$:

$$[\bar{K}]_n\{\Delta u\}_n = \{\Delta F\}_n \tag{13}$$

Because $[\bar{K}]_n$ is a function of $\{\Delta\sigma\}_n$ and $\{\Delta\epsilon\}_n$ the above equations are nonlinear and require iteration to solve. The Newton-Raphson method gives for iteration k:

$$\{\Delta u\}_{n,k} = \{\Delta u\}_{n,k-1} - [J]^{-1}_{n,k-1} \{\Delta\psi\}_{n,k-1} \tag{14}$$

Where the residual vector is:

$$\{\Delta\psi\}_{n,k-1} = [\bar{K}]_{n,k-1} \{\Delta u\}_{n,k-1} - \{\Delta F\}_n \tag{15}$$

The components of the Jacobian $[J]_{n,k-1}$ are found by taking derivatives of eq. (15) with respect to the components of $\{\Delta u\}_n$, in index notation:

$$J_{ij}{}_{n,k-1} = \left.\frac{\partial(\bar{K}_{i\ell}\Delta u_\ell - F_i)}{\partial\Delta u_j}\right|_{n,k-1} = \left[\bar{K}_{ij} + \frac{\partial\bar{K}_{i\ell}}{\partial\Delta u_j}\ \Delta u_\ell\right]_{n,k-1} \tag{16}$$

or

$$[J]_{n,k-1} = [\bar{K}]_{n,k-1} + [\bar{K}]'_{n,k-1}$$

Owen and Hinton (1980) state that in general $[\bar{K}]'$ is not symmetric. In addition, it is relatively difficult and expensive to compute. Based on the classical graphical interpretation of the Newton-Raphson method, Owen and Hinton (1980) suggest that, instead of using eq. (17), the Jacobian be approximated by the tangent stiffness matrix at "n". This requires using only the $[D]_{n,k-1}$ matrices (see paragraph following eq. (7)) for the formation of the system tangent stiffness matrix, call the result $[K]_{n,k-1}$. The matrix $[\bar{K}]_{n,k-1}$ is still needed for the calculation of the residual vector eq. (15). Thus the "tangent stiffness method" consists of iteration using eqs. (14) and (15) with

$$[J]_{n,k-1} \approx [K]_{n,k-1} \tag{18}$$

As an alternative, eq. (17) can be approximated by neglecting the last term†, i.e.

$$[J]_{n,k-1} \simeq [\bar{K}]_{n,k-1} \tag{19}$$

Substituting eqs. (15) and (19) into eq. (14) gives:

$$\{\Delta u\}_{n,k} = \{\Delta u\}_{n,k-1} - [\bar{K}]^{-1}_{n,k-1} \left\{ [\bar{K}]_{n,k-1} \{\Delta u\}_{n,k-1} - \{\Delta F\}_n \right\}$$

or

$$\{\Delta u\}_{n,k} = [\bar{K}]^{-1}_{n,k-1} \{\Delta F\}_n \tag{20}$$

or

$$[\bar{K}]_{n,k-1} \{\Delta u\}_{n,k} = \{\Delta F\}_n \tag{21}$$

Inspecting eq. (21) it is seen that this second approximation to the Newton-Raphson method is one of the possible forms of the classical method of successive approximations, and shall be referred to by that name in the remainder of the report. It can be applied by either using eq. (21) directly or using eqs. (14), (15), and (19). The use of equations (14), (15) and (19) would be expected to be somewhat less susceptible to round-off error; however, no significant differences were detected in the examples analyzed in this study.

The calculation of the residual vector, eq. (15), requires some special attention. One option is, at the element level, to use $[\bar{D}]_{n,k-1}$ to calculate an element stiffness matrix, to then multiply this matrix by the k-1 estimate of the displacements for the nodes defining the element, and to add it to the negative of the element load matrix (i.e., a direct evaluation of eq. (15) at the element level). The resulting element

---

† When convergence occurs, eq. (13) is exactly satisfied. Thus, neither this approximation, nor the one leading to the tangent stiffness method, has any effect on the final accuracy of the solution.

residual vectors are then combined in the usual way to give the system residual vector. While this operation requires little additional computational effort for the method of successive approximations, it does for the tangent stiffness method. The cause of this additional work is the need to calculate two element stiffness matrices† based upon the two quantities $[D]_{n,k-1}$ and $[\bar{D}]_{n,k-1}$. The first element stiffness matrix is needed in the calculation of the system tangent stiffness matrix, and the second in the calculation of the residual vector. An alternative procedure for the calculation of the residual vector avoids this additional effort; however, it places certain restrictions on the order of the integration used in establishing the element matrices. In this second approach, an initial stress vector is calculated:

$$\{\Delta\sigma_o\}_{n,k-1} = [\bar{D}]_{n,k-1} \{\Delta\varepsilon\}_{n,k-1} \tag{22}$$

The following incremental stress-strain equation is then used in the calculation of the element stiffness and load matrices:

$$\{\Delta\sigma\}_{n,k} = [D]_{n,k-1} \{\Delta\varepsilon\}_{n,k} + \{\Delta\sigma_o\}_{n,k-1} \tag{23}$$

Assuming that all numerical integrations are done with the same accuracy, it is easy to show that the use of eq. (23) yields element matrices that, when combined at the system level, give the desired tangent stiffness matrix and the residual force vector. In this operation care must be taken to ensure that all numerical integrations are of the same order. For example, for a four node element, if four point integration is used in calculating the element matrices but the incremental properties (eq. (23)) are calculated only at the element center (i.e., assumed constant over the element),

---

† Geometric quantities such as the shape function derivates need only be calculated once.

an inconsistency† exists and convergence is drastically affected. Thus when using this scheme the stresses, strains and properties must be evaluated at the integration points. However, for a four node element it has been shown (see Herrmann 1972) that the stress calculations are most accurate at the element center, not at the integration points. For this reason, and because the four node element is poorly behaved for undrained conditions, the use of an eight or nine node element with stresses and properties calculated at each of the four quadrature points is recommended for future work.

3.5.2. Reduction Of The Number Of Equation Triangularizations

The use of the "modified" Newton method (referred to as the "initial stiffness" method by Owen and Hinton 1980) is the classical means for reducing the number of triangularizations of the left-hand side of the simultaneous equations. In this procedure, the stiffness matrix is updated only occasionally. Because of the drastic difference in stiffness, in elasto-plasticity problems (which is encountered in a progression from elastic behavior to yielding and from loading to unloading) it is desirable to update at least once each increment. For this study, the stiffness matrix is updated in the second iteration of each increment. The second iteration was chosen because it often takes at least one iteration to establish the loading-unloading characteristics for the increment. In addition, it is updated in the first iteration of the first increment of each new loading segment (a loading segment generally consists of many increments). This update is done because there is often a switch from loading to unloading at the beginning of a new loading segment. Finally, the stiffness matrix is updated every IRPET iteration, where IRPET is specified by the user. The use of the "modified Newton" procedure in conjunction with the method of successive approximations,

---

† That is, the residual node point forces are not accurately made to be zero.

implemented by means of eq. (21) instead of eq. (14), requires that the difference between the present stiffness matrix and the last stiffness matrix to be triangularized, be accounted for. With this end in mind, eq. (21) is written in the form:

$$[\bar{K}]_L \{\Delta u\}_{n,k} = \{\Delta F\}_n - \left\{[\bar{K}]_{n,k-1} - [\bar{K}]_L\right\} \{\Delta u\}_{n,k} \tag{24}$$

Now if the quantity $\{\Delta u\}_{n,k}$ on the right-hand side is estimated by means of $\{\Delta u\}_{n,k-1}$ (as convergence occurs no approximation is introduced):

$$[\bar{K}]_L \{\Delta u\}_{n,k} = \{\Delta F\}^*_{n,k-1} \tag{25}$$

where

$$\{\Delta F\}^*_{n,k-1} = \{\Delta F\}_n - \left\{[\bar{K}]_{n,k-1} - [\bar{K}]_L\right\} \{\Delta u\}_{n,k-1} \tag{26}$$

the last term on the right-hand side of eq. (26) can be easily evaluated by forming a pseudo initial stress vector:

$$\{\Delta \sigma_o\}_{n,k-1} = \left\{[\bar{D}]_{n,k-1} - [\bar{D}]_L\right\} \{\Delta \varepsilon\}_{n,k-1} \tag{27}$$

This pseudo initial stress vector contributes to the element load matrix in the usual way. Because this quantity approaches zero as convergence occurs, it is not necessary to evaluate the stress-strain properties at the quadrature points as is the case in calculating the residual vector for the tangent stiffness method (see previous discussion).† The inclusion of a similar term for the tangent stiffness method and for successive approximations, evaluated by means of eq. (14), appears not to be standard practice, and was not considered in this study.

---

† Although the evaluation of the properties at the quadrature points is not necessary to assure convergence, it might improve the rate of convergence; this possibility was not explored in the study.

Currently, one of the most popular means for reducing the number of triangularizations is to use a quasi-Newton update of the triangularized or inverted stiffness (approximate Jacobian) matrix. As noted previously, the only quasi-Newton methods considered in this study are those that directly update the triangularized form of the matrix and do not disturb its banded nature. In order to satisfy the requirement of ease of implementation, a simpler quasi-Newton update formula than those available in the literature was sought. Because it is generally agreed that the BFGS update formula is the best available (see Fletcher 1980), a formula of similar form was desired. Denoting the triangularized form of the tangent stiffness matrix as [K]*, the update for the "k" iteration expressed in index form is:

$$K^*_{ij_k} = K^*_{ij_{k-1}} + a_i \Delta^*_j - b_i \sum_{\ell=i}^{j} \delta_\ell K^*_{\ell j_{k-1}} \tag{28}$$

where

$$a_i = \frac{\Delta^*_i}{\sum_{\ell=i}^{I} \Delta^*_\ell \delta_\ell} \tag{29}$$

$$b_i = \frac{\sum_{\ell=i}^{I} K^*_{I\ell_{k-1}} \delta_\ell}{\sum_{m=i}^{I} \left\{ \delta_m \sum_{\ell=i}^{m} K^*_{\ell m_{k-1}} \delta_\ell \right\}} \tag{30}$$

$$I = i + \text{bandwidth} \tag{31}$$

The vector $\{\Delta\}^*$ is the result of a block formulation of the residual vector $\{\psi\}_{n,k-1}$ and its reduction up to and including row "i". The results of the previous iteration are denoted by $\{\delta\} = \{\Delta u\}_{n,k-1} - \{\Delta u\}_{n,k-2}$. When eqs. (29) and (30) are substituted into eq. (28) it is easily seen that the update formula has a form similar to the BFGS formula (for the Jacobian not its inverse). It is also easy to verify that the update formula satisfies the "quasi-Newton condition" (see Geradin et al 1981). The

implementation of eq. (28) is straightforward, with the updating of K* proceeding simultaneously with the reduction of the current residual vector.

3.5.3 Convergence Acceleration

The acceleration of convergence and enlargement of the radius of convergence of an iterative scheme by means of some type of extrapolation procedure can often be very cost effective. The most convenient way of expressing such an acceleration is in terms of a convergence (iteration, relaxation) factor; i.e., an improved estimate of the solution vector $\{\Delta u\}^*_k$, for iteration k, is expressed in terms of the estimate obtained from the solution procedure $\{\Delta u\}_k$, the previous estimate $\{\Delta u\}_{k-1}$ and a convergence factor C. For component "i" this extrapolation is expressed in the form:

$$\Delta u^*_{i_k} = \Delta u_{i_{k-1}} + C_{i_k} [\Delta u_{i_k} - \Delta u_{i_{k-1}}] \tag{32}$$

Written in this form, a value of C>1 leads to "over-relaxation," C<1 to "under-relaxation" and C=1 to no extrapolation. The simplest approach is to use a constant acceleration factor $C_{i_k} = C_o$, for all iterations and all components of the solution vector, and to require the user to supply the value based on past experience. A more successful approach is to use some rational criterion to calculate near optimum values of the factor (see Isaacson and Keller, 1966). For single degree of freedom problems (unknown x), the most popular procedure is the Aitken's ∇^2 method, which yields extrapolated estimates for odd iterations beginning with the 3rd, i.e.,

$$x^*_k = x_k - \frac{(x_k - x_{k-1})^2}{x_k - 2x_{k-1} + x_{k-2}} \qquad k=3,5,7... \tag{33}$$

When this equation is expressed in the form of eq. (32) it gives a value for the convergence factor of:

$$C_k = \begin{cases} 1.0 & k=1,2,4,6... \\ \dfrac{x_{k-1} - x_{k-2}}{-x_{k-2} + 2x_{k-1} - x_k} & k=3,5,7,... \end{cases} \quad (34)$$

In order to avoid possible numerical problems when one is either far away from or very near to the solution, it is desirable to place limits on the value calculated by eq. (34). For the purpose of this study, such limits are expressed in the form

$$\frac{1}{C_L} \leq C \leq C_L \quad (35)$$

where $C_L \geq 1$ and is user supplied.

Eq. (34) applies to a single variable; the question is how to extend it to a multi-degree of freedom problem. Two schemes are proposed. In the first, for a given iteration, a constant value of C_k is used for all components of $\{\Delta u\}_k$. The value of C_K is calculated by using the norms ($N_k = \Sigma_i |\Delta u_{i_k}|$) of the solution vector in eq. (34). This procedure is based on the assumption that the convergence characteristics of all the components of the vector are similar. In the second procedure, eq. (34) is applied to each component of $\{\Delta u\}$ to give a different convergence factor for each component, for each iteration.

3.5.4 Accounting For The Inconsistency In The Incremental Stress-Strain Relation

As previously noted, until convergence occurs, the estimates of $\{\Delta\epsilon\}_{n,k-1}$ and $\{\Delta\sigma\}_{n,k-1}$ used in the calculation of $[\bar{D}]_{n,k-1}$ do not in fact satisfy eq. (6). Three methods for handling this inconsistency are explored.

Because in fact the inconsistency disappears as global convergence occurs (i.e., as $[\bar{D}]_{n,k-1} \approx [\bar{D}]_{n,k-2}$), the first alternative is to do nothing; i.e., to rely completely on global iteration.

In the second approach, local iteration is introduced in the calculation of $[\bar{D}]$ so as to remove the inconsistency (see Herrmann and Taylor 1974). Using $\{\Delta\epsilon\}_{n,k-1}$

and $\{\Delta\sigma\}_{n,k-1}$, $[\bar{D}]_{n,k-1}$ is calculated. The values of $\{\Delta\epsilon\}_{n,k-1}$ and $[\bar{D}]_{n,k-1}$ are then used in conjunction with eq. (6) to calculate a new estimate of stress $\{\Delta\sigma\}^*_{n,k-1}$ which is in turn used with $\{\Delta\epsilon\}_{n,k-1}$ to calculate a new estimate for the incremental properties $[\bar{D}]^*_{n,k-1}$. This process is continued until convergence is achieved for $\{\Delta\sigma\}^*_{n,k-1}$. Because of the global iteration which also tends to remove the inconsistency, the convergence limit on the local iteration is taken to be only 1/10 as restrictive as the global requirement. The stress estimate is iteratively modified (instead of the strain estimate) in order to maintain a compatible global displacement field as required by the admissibility conditions of the finite element procedure.† The introduction of local iteration (for all points where the incremental stress-strain properties are required) of course substantially increases, in a given iteration, the number of calls to subroutine CLAY, presumably with a corresponding reduction in the number of global iterations.

In the third approach, the inconsistency was expressed in terms of a pseudo-residual stress vector, i.e. $\{\sigma_o\}_{n,k-1} = \{\sigma\}_{n,k-1} - [\bar{D}]_{n,k-1}\{\Delta\epsilon\}_{n,k-1}$, which was then incorporated into the system residual stress vector.

3.6 Finite Element Implementation

Following the instructions given in Herrmann et al (1981), subroutine CLAY was installed in a standard two-dimensional, four node, nonlinear finite element program (NTD). The program was extended beyond its original successive approximation capability to include the several options outlined in Section 3.5. The program was used for the numerical study described in the following section. After the completion of the evaluation, in order to somewhat simplify the code, two of the less robust features were removed (see discussion in next section). A listing and brief user's manual for the code are given in Appendix B.

---

† The actual consequences of modifying the strain estimate were not studied.

The original NTD code was relatively efficient, uncluttered and well documented. However, once the several options described above were included, these characteristics were lost. The problem is that the features described in Section 3.5, when taken in all possible combinations, lead to a total of 72 different solution strategies; hence, the flow through the program has become rather complex. Thus, while the code given in the Appendix should be of considerable value in possible future extensions of the numerical study, it is not recommended for extensive production applications. Rather, it is recommended that, in a future project, a production code with far fewer options be developed.

4. NUMERICAL STUDY

4.1 Scope Of Study

The purposes of the project were to compare the effectiveness of various numerical strategies for implementing, in finite element analyses, the bounding surface plasticity model for cohesive soils, and to investigate the numerical characteristics of the model. While theoretical considerations can lead to general statements concerning convergence, etc., many subtle differences can only be determined by numerical experimentation. The theoretical foundations of the methods being investigated are for the most part well established, and thus this study concentrated on numerical experimentation. Three representative problems were selected for study and analyzed by the several different solution strategies described in the previous section. Results of the several analyses were compared on the bases of reliability and computation efficiency.

4.2 Selection Of Representative Problems

The criteria for selecting the problems were that they a) be related to actual geotechnical engineering problems, b) be numerically challenging and c) be simple enough that the overall cost of the study would not exceed the available resources.

The soil properties (model parameters) used in the study were found by applying the calibration procedure of DeNatale (1982) to the test results reported by Jafroudi (1982) (see also Herrmann et al 1981); the properties are given in Table 1. The degree of initial overconsolidation (as a function of depth in the soil mass) was varied among the several analyses. Convergence of the solution was determined by requiring that:

$$\frac{\sum_i |\Delta u_{i_k} - \Delta u_{i_{k-1}}|}{\sum_i |\Delta u_{i_k}|} \leq \text{ERLMT} \tag{36}$$

A value of ERLMT = .001 was found to be adequate for most cases.

The first problem, studied in Figure 8, was a simple triaxial test requiring only one element for modeling. Several hundred analyses were performed. All solution strategies were used, a number of different loading histories were evaluated and several different initial overconsolidation ratios were considered. Because of the simplicity of the problem, only limited conclusions could be drawn from a comparison of the results.

The second problem consisted of the vertical loading of a strip footing resting on a clay deposit. The grid used in the analysis is shown in Figure 9. The extent of the clay deposit was limited in size in order to keep the computer cost of an individual analysis small.† Analyses were performed for normally, moderately and highly overconsolidated soils. For normally and moderately overconsolidated soils, the model is numerically so well behaved that little distinction could be made between the several solution strategies. Thus, the bulk of the analyses (18 in number) were for a highly overconsolidated soil. However, even under these conditions, the problem

---

† An analysis of a footing on a highly overconsolidated soil in which the load-deformation curve was taken beyond the maximum load required from 10-15 increments and cost on the order of $5-$10.

was not sufficiently challenging to reveal major differences among the several solution schemes (with one exception which is discussed in the following section).

The third problem consisted of a highly overconsolidated clay deposit supported by a rigid retaining wall that experienced rotation about its base (see Figure 8), thus leading to a significant reduction in confining pressure adjacent to the wall. This problem was found to be more challenging and served to distinguish among the several solution strategies. A total of 33 different analyses were performed on the problem.† The grid used for the majority of these analyses is shown in Figure 9. The selection of the grid, the convergence criterion and the solution increment size, required to achieve accurate results involved performing analyses with both coarser and finer grids, larger and smaller solution steps and different convergence criteria.

The conclusions drawn from a comparison of the results of the several analyses of the three representative problems are discussed in the next section.

4.3 Comparison Of Results

The only definite conclusion that could be drawn from problem #1 was that the proposal to treat the inconsistency in the incremental stress-strain law by calculation of a pseudo-residual stress was entirely unsatisfactory (convergence could not be achieved), and was thus abandoned for the remainder of the study. A postmortem investigation of the proposed method revealed no theoretical justification for its use, and thus its failure is not surprising.

While the second problem revealed some differences among the several solution strategies, on the whole, convergence occurred so quickly that, with one exception, no definitive conclusions could be drawn. The one conclusion from the second problem

---

† In addition, a significant number of incorrect runs were made prior to the detection of the numerical integration problem for the residual vector described in a previous section.

was that the quasi-Newton scheme is not nearly as robust as the modified Newton procedure, and at that point in the study it was dropped from further consideration. At this stage in the development of quasi-Newton methods, it is concluded that the several advantages claimed for optimization and nonlinear elasticity problems do not carry over to plasticity problems, at least as far as the bounding surface model and the scope of this study are concerned. Other trends noted in the second problem were more clearly evident in the third example and are discussed in conjunction with that study.

Among the analyses performed on the third problem, 24 used the same grid (Figure 8), solution increment size and convergence criterion and thus can be directly compared. These 24 solutions do not completely exhaust the 32 solution strategies remaining within the scope of the study after eliminating the two components discussed in the previous paragraphs (the quasi-Newton scheme and the pseudo-residual stress representation of the inconsistency in the incremental stress-strain law).† It is felt, however, that these 24 cases sufficiently span the strategy space to be representative and permit definitive conclusions to be drawn. The characteristics of these analyses are described in Table 2. In the last column of the table a measure of the cost of each analysis is given. In the calculation of this measure, the actual cost of the analysis was slightly modified to account for anticipated future savings due to improved efficiencies of the equation solver and of subroutine CLAY, and, in addition, the pre-and post-processing costs are not included. Because of differences in coding practices, in relative costs of computation and storage for various computers, etc., these figures contain a degree of uncertainty, and thus differences of less than 25%-50%

---

† If consideration is given to the considerable latitude that exists in assigning values to C_L (eq. (35)), IREPT (eq. (25)), the step size and the ratio of the local to global convergence criteria it is seen that the actual number of possibilities is really far greater than 32.

probably are not significant. The lack of convergence, noted in the table, for several of the analyses was due in some cases (those with a high computational cost measure) to the limit set on the maximum number of iterations in a given increment and in others (those with an indicate low cost measure) to the satisfaction of the convergence criterion, eq. (36), even though convergence had not occurred. In either case, a lack of robustness is indicated. In the following paragraphs the conclusions drawn from comparing the results of the several analyses performed in this study are discussed.

One proposal often made for nonlinear inelasticity problems is to use small increment sizes to avoid excessive (all) iteration. The results of this study do not support this suggestion. In Figure 10 the relative costs required to reach a certain point in the solution history is plotted versus the ratio of the number of steps used, to the minimum number required to accurately reach the point. This plot clearly suggests, from a computational efficiency standpoint, that one should use the largest step size that will give acceptable accuracy. (If the step size is made too large, the numerical integration error in eq. (6) becomes unacceptable.)

The methods of successive approximations and tangent stiffness exhibit very similar characteristics with no clear cut difference between them; this conclusion runs contrary to the suggestion of Owen and Hinton (1980) of the superiority of the tangent stiffness method. Both methods, if properly supplemented with other components of the solution strategy, are quite robust, economical and are sufficiently accurate for solving bounding surface elasto-plasticity problems in geotechnical engineerng. The method of successive approximations has some slight, but not major advantages, in ease of implementation.

The use of local iteration considerably improves the reliability of the convergence criterion and in most cases improves the efficiency of the analysis.

The consequences of using the modified-Newton scheme and/or an acceleration factor are somewhat interrelated and hard to separate.† When the modified Newton scheme is not employed, the use of a variable acceleration factor is of major benefit for the method of successive approximations and of lesser value for the tangent stiffness method. When used in conjunction with the modified Newton scheme, a variable acceleration factor has some value for the tangent stiffness method, but appears to offer little advantage for successive approximations. Of the two schemes tested for calculating a variable acceleration factor, the one that differs from component to component is preferable. The modified Newton method and the calculation of a variable convergence factor are both very simple operations to implement.

For the grids used in this study, the percentage of the total solution cost spent in evaluating subroutine CLAY ranged from about 20% to 60%. For the larger grids needed for production problems, it is anticipated that the costs would be of the order of 10%-20%.

5. CONCLUSIONS

Based on the comparisons made in this study several conclusions are drawn. To what extent these conclusions are generally applicable to very different geotechnical problems and to other plasticity models is unknown.

The bounding surface model for cohesive soils is simple to implement in a standard, nonlinear finite element analysis using either successive approximations or

---

† This interrelationship is apparently due to the fact that only occasionally updating the stiffness matrix has somewhat of a disruptive effect on the extrapolation scheme used in the acceleration of convergence. This suggests that results obtained from iterations involving updating and those not involving updating should not be used together in eq. (34).

the tangent stiffness method. It is numerically well behaved and does not lead to prohibitive computational costs.

Both the methods of successive approximations and tangent stiffness can be viewed as approximations to the Newton-Raphson method, and there is little to choose between them. Either method would appear suitable for a production finite element program for geotechnical problems. Successive approximations is somewhat easier to implement and somewhat more efficient; however, it is slightly less robust than the tangent stiffness method.

The introduction of local iteration in the calculation of the incremental stress-strain properties is desirable, and will in the future be incorporated into subroutine CLAY so as not to clutter the logic of the parent finite element program.

Finally, it is recommended that a production program incorporate provisions for a modified Newton analysis and for a variable convergence factor based on the use of Aitken's ∇^2 method applied to each component of the solution vector. Both of these procedures are simple to implement.

It is concluded that the effective use of a quasi-Newton method for elasto-plasticity problems will need to await the development of simple but robust updating formulas for the banded upper triangularized form of the stiffness matrix.

6. RECOMMENDED FUTURE WORK

Two major components of the overall project remain; i.e., the development of one or more production finite element programs for geotechnical engineering problems, and a verification study for actual field structures and/or large centrifuge models. The first of the remaining components is further discussed in the following paragraphs.

Prior to any production program development, subroutine CLAY should be recoded in order to improve its efficiency and portability. It is then recommended that two production programs for geotechnical engineering problems be developed. The first

would be two-dimensional and applicable to either drained or undrained conditions. It would contain simple pre-processing routines for grid generation and the description of initial stress and saturation states. The program would be an outgrowth of the program used in the present study. The grid generation scheme and the equation solver would be replaced by recently developed optimized versions. The type of element would be carefully selected, in light of accuracy requirements for undrained conditions.

The second program would be three-dimensional and, in addition to being applicable to ideal drained and undrained conditions, would consider the effects of partial saturation and the movement of pore water under saturated and nonhomogeneous stress state conditions. Initially, a frontal equation solver would be used; time permitting the new iteration scheme being developed by Professor Taylor at UCB would be considered as an alternative.

As time permits, special features such as bending elements, incremental excavation and construction options, more elaborate pre and post-processing schemes and complete dynamic dimensioning (as opposed to the partial dynamic dimensioning used in the current two-dimensional program) would be included in the programs. Carefully documented user manuals would be prepared for both programs.

REFERENCES

1. Ajaz, A. and Parry, R.H.G. (1976), "Bending Test for Compacted Clays," Journal of Geotechnical Engineering Division, ASCE, v. 102, no. GT9, pp. 929-943.

2. Anderson, K.H. and Stenhamar, P. (1982), "Static Plate Loading Tests on Overconsolidated Clay," Journal of the Geotechnical Engineering Division, ASCE, v. 108, no. GT7, pp. 918-934.

3. Balasubramian, A.S., Sivandran, C. and Ho, Y.M. (1979), "Stability and Settlement of Embankments on Soft Bangkok Clay," Proceedings of the Third International Conference on Numerical Methods in Geomechanics, v. 3, pp. 1373-1411.

4. DeNatale, J.S. (1982), "On the Calibration of Constitutive Models by Multivariate Optimization. A Case Study: The Bounding Surface Plasticity Model," Ph.D. Thesis, Department of Civil Engineering, University of California, Davis.

5. Desai, C.S., Phan, H.V. and Sture, S. (1981), "Procedure, Selection and Application of Plasticity Models for a Soil," International Journal for Numerical and Analytical Methods in Geomechanics, v. 5, pp. 295-311.

6. Fletcher, R. (1980), Practical Methods of Optimization, Vol. 1: Unconstrained Optimization, John Wiley and Sons, Chichester, U.K.

7. Geradin, M., Idelsohn, S. and Hogge, M. (1980), "Nonlinear Structural Dynamics via Newton and Quasi-Newton Methods," Nuclear Engineering Design, v. 54, pp. 339-348.

8. Geradin, M., Idelsohn, S. and Hogge, M. (1981), "Computational Strategies for the Solution of Large Nonlinear Problems via Quasi-Newton Methods," Computers and Structures, v. 13, pp. 73-81.

9. Hanzawa, H. (1979), "Undrained Strength Characteristics of an Alluvial Marine Clay in the Tokyo Bay," Soils and Foundations, v. 19, no. 4, pp. 69-84.

10. Herrmann, L.R. (1965), "Elasticity Equations for Incompressible and Nearly Incompressible Materials by a Variational Theorem," AIAA Journal, v. 3, no. 10, pp. 1896-1900.

11. Herrmann, L.R. (1972), "Interpretation of Finite Element Procedure as Stress Error Minimization Procedure," Journal of the Engineering Mechanics Division, ASCE, v. 98, no. EM5, pp. 1330-1336.

12. Herrmann, L.R. and Taylor, M.A. (1974), "Characterization of the Structural Behavior of Rock Masses," Final Report to the U.S. Bureau of Mines, Contract #G0133122, v. I, Department of Civil Engineering, University of California, Davis.

13. Herrmann, L.R., Dafalias, Y.F. and DeNatale, J.S. (1980), "Bounding Surface Plasticity for Soil Modeling," Department of Civil Engineering, University of California, Davis, Final Report to the Civil Engineering Laboratory of the Naval Construction Battalion Center, Port Hueneme, CA, Order No. USN N62583-80-M-R478.

14. Herrmann, L.R., DeNatale, J.S. and Dafalias, Y.F. (1981), "Numerical Evaluation of a Bounding Surface Plasticity Model for Clays (interim report)," Department of Civil Engineering, University of California, Davis.

15. Herrmann, L.R., Shen, C.K., Jafroudi, S., DeNatale, J.S. and Dafalias, Y.F. (1981b), "A Verification Study for the Bounding Surface Plasticity Model for Cohesive Soils," Department of Civil Engineering, University of California, Davis, Final Report to the Civil Engineering Laboratory of the Naval Construction Battalion Center, Port Hueneme, CA, Order No. N62583-81-M-R320.

16. Herrmann, L.R., Dafalias, Y.F. and DeNatale, J.S. (1982), "Numerical Implementation of a Bounding Surface Soil Plasticity Model," Proceedings of the International Symposium on Numerical Models in Geomechanics (in press).

17. Isaacson, E. and Keller, H.B. (1966), Analysis of Numerical Methods, John Wiley and Sons, New York.

18. Jafroudi, S. (1982), "Experimental Verification of a Bounding Surface Plasticity Theory for Cohesive Soil," Ph.D. Thesis, Department of Civil Engineering, University of California, Davis (to appear).

19. Mathies, H. and Strang, G. (1979), "The Solution of Nonlinear Finite Element Applications," International Journal of Numerical Methods in Engineering, v. 14, pp. 1613-1626.

20. Nagtegaal, J.C., Parks, D.M. and Rice, J.R. (1974), "On Numerically Accurate Finite Element Solutions in the Fully Plastic Range," Comp. Meth. Appl. Mech. Eng., v. 4, pp. 153-178.

21. Owen, D.R.J. and Hinton, E. (1980), Finite Elements in Plasticity - Theory and Practice, Pineridge Press, Swansea, U.K.

22. Radhakrishnan, N. and Reese, L.C. (1969), "Behavior of Strip Footings on Layered Cohesive Soils," Proceedings of the Symposium on Application of Finite Element Methods in Civil Engineering, ASCE, pp. 691-728.

23. Sandhu, R.S. and Wilson, E.L. (1969), "Finite Element Analysis of Seepage in Elastic Media," Journal of the Engineering Mechanics Division, ASCE, v. 95, no. EM3, pp. 641-651.

24. Sangrey, D.A., Henkel, D.J. and Epsig, M.I. (1969), "The Effective Stress Response of a Saturated Clay Soil to Repeated Loading," Canadian Geotechnical Journal, v. 6, no. 3, p. 241-252.

25. Schubert, L.K. (1970), "Modification of a Quasi-Newton Method for Nonlinear Equations with a Sparse Jacobian," Mathematics of Computation, v. 24, pp. 27-30.

26. Zienkiewicz, O.C. (1977), The Finite Element Method, 3rd edition, McGraw-Hill, London, U.K.

27. Zienkiewicz, O.C., Taylor, R.L. and Baynham, J.M.W. (1981), "Mixed and Irreducible Formulations in Finite Element Analysis," Proceedings of the International Symposium on Hybrid and Mixed Finite Element Methods, Atlanta, GA.

Table 1: Results of the Manual and Computer Aided Model Calibration Procedures as Applied to the Experimental Data of Jafroudi (1982).

Parameter[1]	Manual Solution[2]	Automated Solution[3]
λ^*	0.130	0.130
κ^*	0.018	0.018
M_c^*	1.18	1.18
M_e^*	0.87	0.87
ν,G	0.30	5900
P_ℓ	100	100
R_c	2.40	2.509
R_e	2.25	2.246
A_c	0.04	0.031
A_e	0.04	0.034
T	0.10	0.046
c	0.71	0.453
s	1.00	1.000
h_c	2.00	0.621
h_e	4.00	0.855
m_c^*	0.20	0.200
m_e^*	0.20	0.200

1 a * indicates that this parameter was assigned a fixed value prior to using the computer aided solution

2 following discovery of an error in the reduction of the laboratory results, the model was recalibrated manually, and reported earlier by Herrmann et al 1981b.

3 the objective function was created with the options set at PLIM = TLIM = 0.20. All data was weighted equally.

Table 2: Comparison of Numerical Results

Run No.	Solution Method		Modified Newton		Acceleration Factor					Local iteration		Convergence occurred within prescribed limits		Average number per increment			Measure of Cost
						Yes											
							Variable Value										
	Successive Approx.	Tangent Stiff.	No	Yes-value of IREPT	No	Const. Value	Norms	Comp.	Value of C_L	No	Yes	No (solution within 5% of correct values)	Yes	Triangular-izations	Back-substitutions	Calls to CLAY	
3	X		X		X						X	X		>16	>16	>2983	>2.72
15		X	X		X						X		X	9.5	9.5	2281	1.92
4	X			10	X						X	X		>2.5	>16	>3161	>2.02
6	X			5	X						X		X	2	6.5	1845	1.20
34		X		5	X						X		X	2.5	10.5	2512	1.62
7	X		X			.7					X	X		>7.5	>7.5	>2785	>2.57
8	X		X			1.3					X	X		>16	>16	>3093	>2.81
9	X		X				X		1.5		X		X	5	5	1185	1.00
33		X	X				X		1.5		X		X	9	9	2282	1.89
10	X		X					X	1.5		X		X	5	5	1182	1.00
32		X	X					X	1.5		X		X	7	7	1905	1.54
24	X			5				X	1.5		X		X	2	5.5	1348	1.20
36		X		5				X	1.5		X		X	2	7.5	2025	1.30
23	X			5					1.5		X		X	2	7.5	1824	1.19
5	X		X		X					X		X		>21	>21	>1344	>2.13
14		X	X		X					X		X		>21	>21	>1344	>2.13
13	X			5	X					X		X		>3	>11.5	>736	>.65
35		X		5	X					X		X		>4.5	>21	>1344	>3.06
19	X			5			X		1.5	X		X		>2.5	>10	>640	>.56
20	X			5				X	1.5	X		X		>3	>11	>704	>0.63
37		X		5				X	1.5	X		X		>4	>18.5	>1184	>0.97
2	X			10	X					X		X		>2	>11.5	>736	>0.58
30		X	X					X	1.5	X		X		>21	>21	>1344	>2.13
31		X	X				X		1.5	X		X		>18.5	>18.5	>1184	>1.87

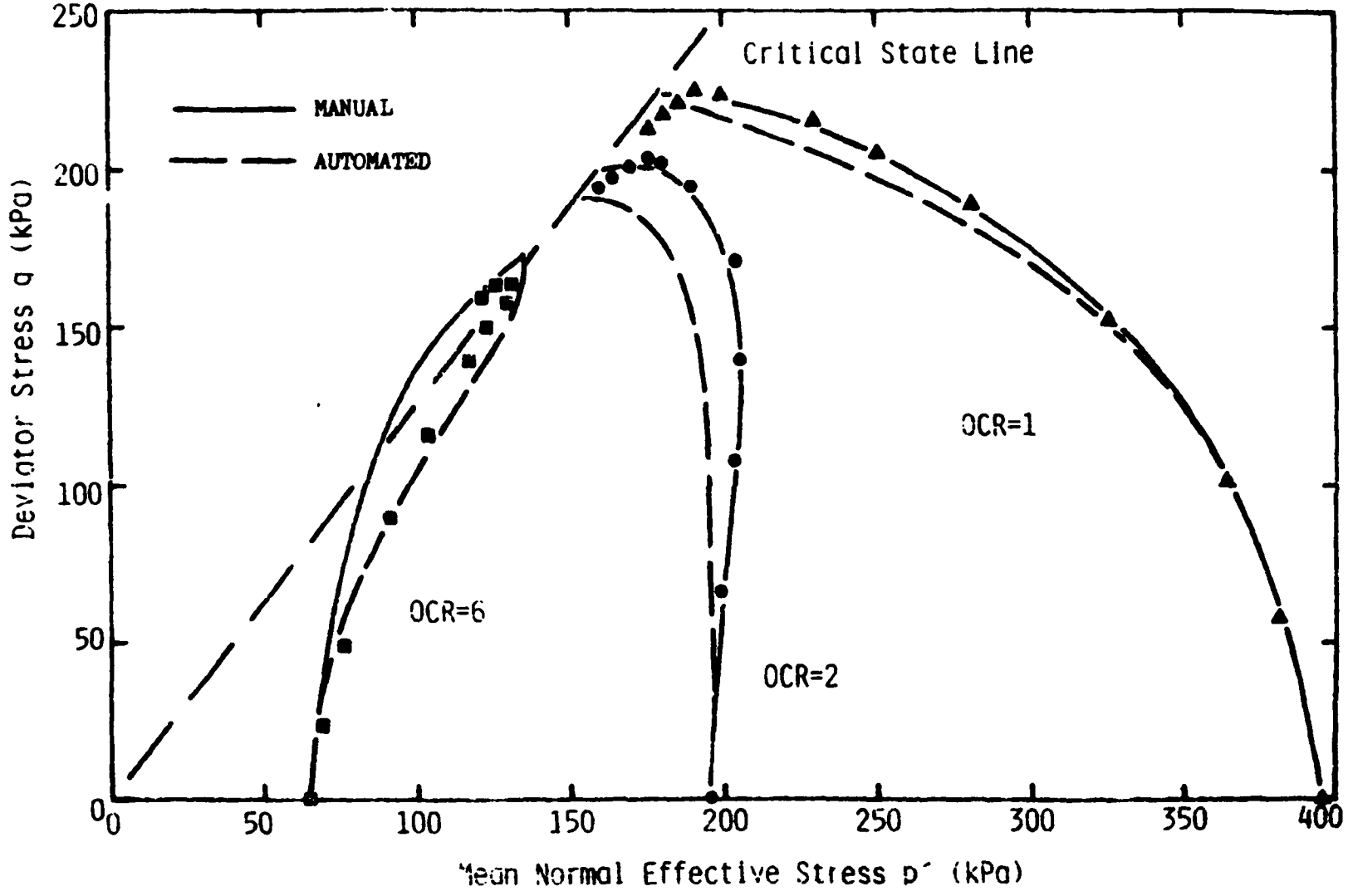


Figure 1: Calibration Curves for q vs. p' Relation in Triaxial Compression

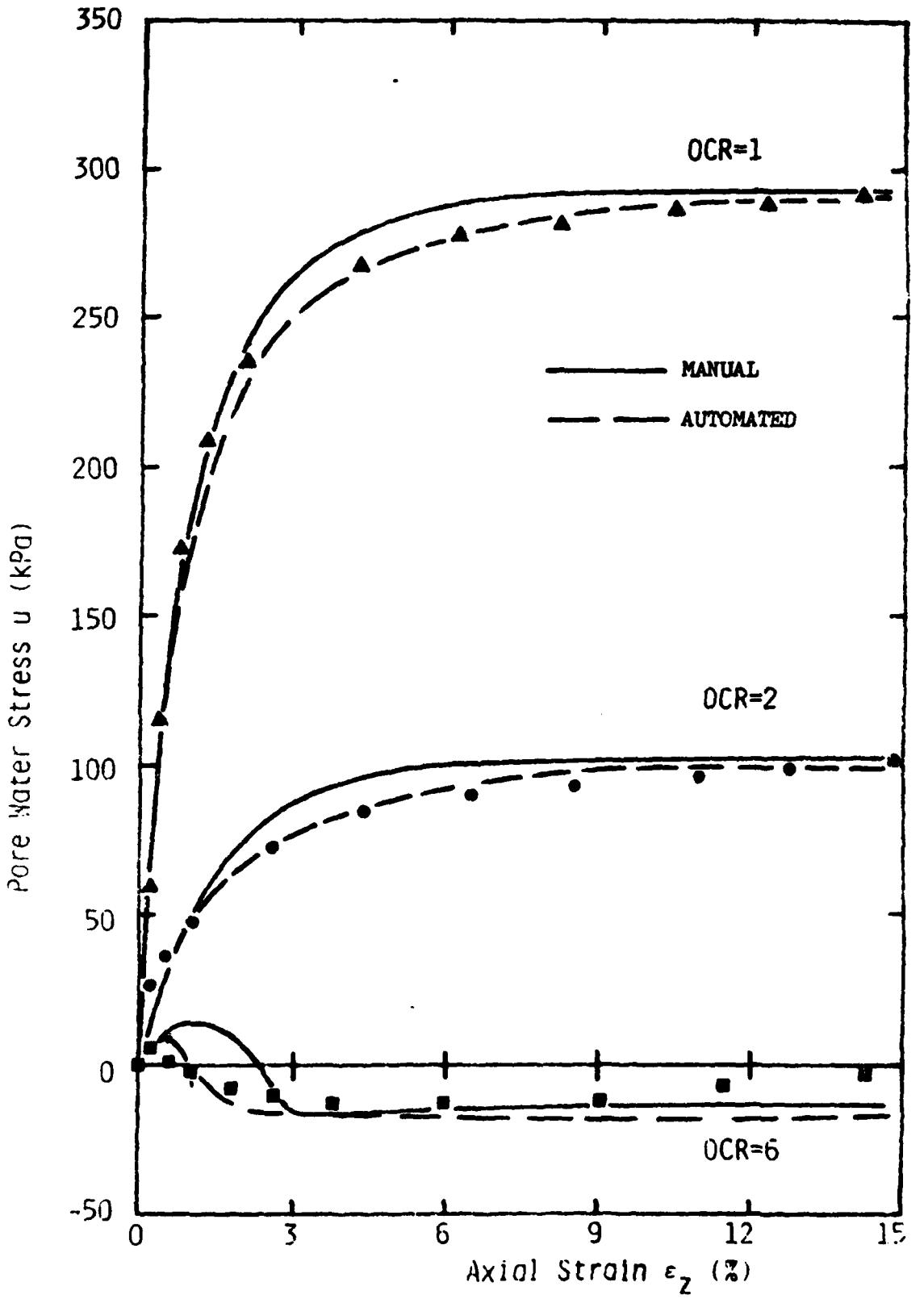


Figure 2: Calibration Curves for σ vs. u Relation in Triaxial Compression

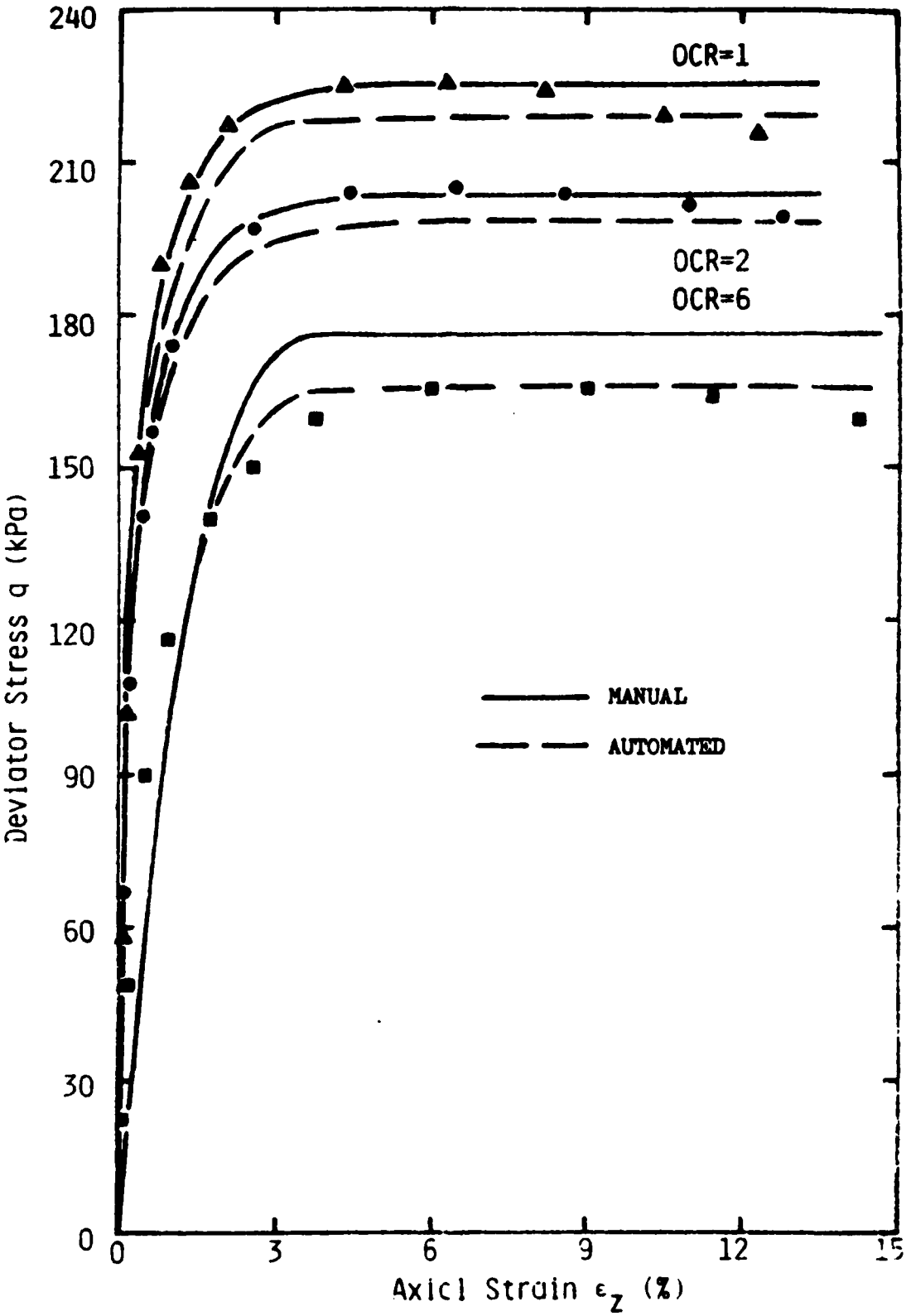


Figure 3: Calibration Curves for q vs. ϵ_z Relation in Triaxial Compression

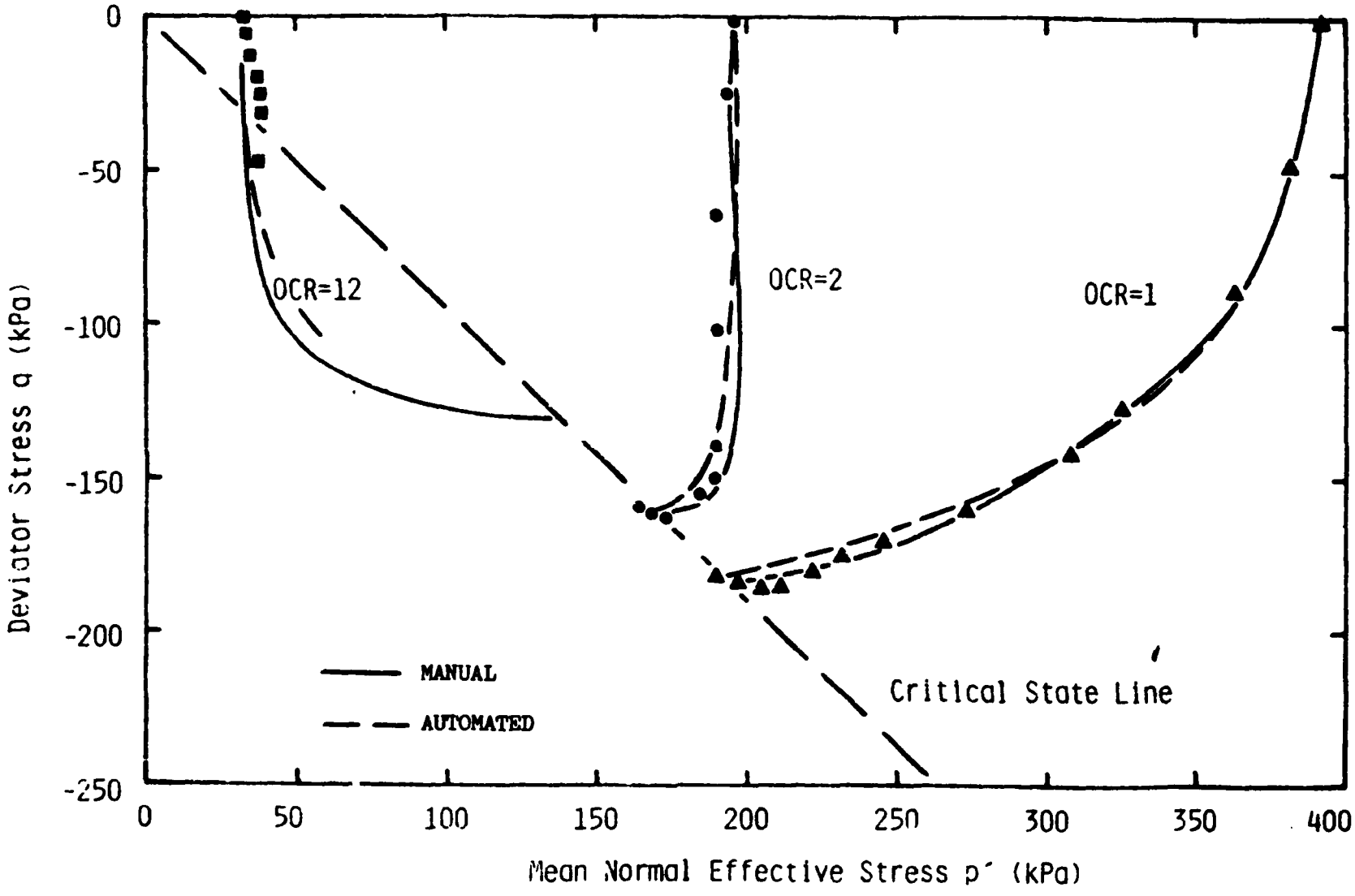


Figure 4: Calibration Curves for q vs. p´ Relation in Triaxial Extension

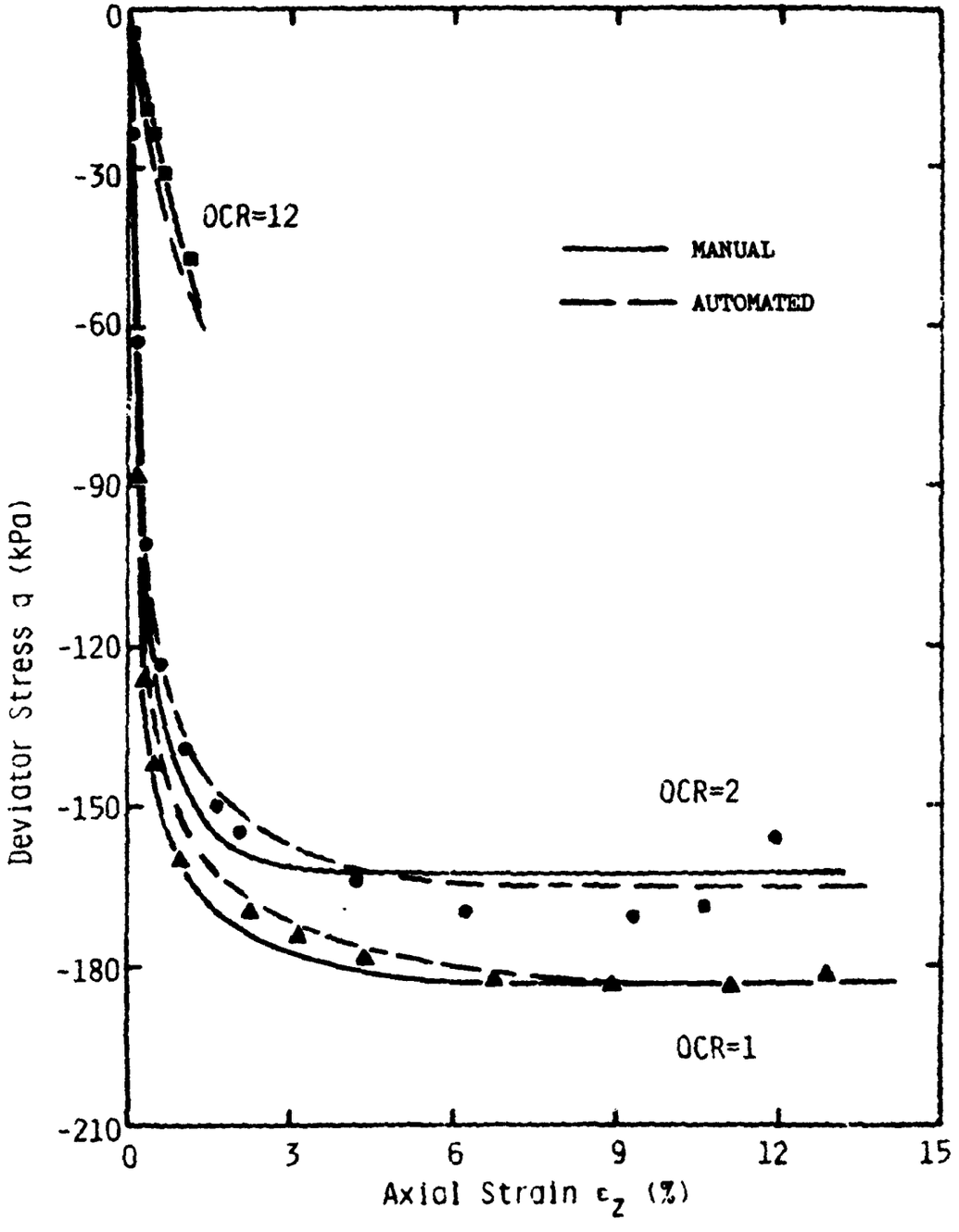


Figure 5: Calibration Curves for q vs. ε_z Relation in Triaxial Extension

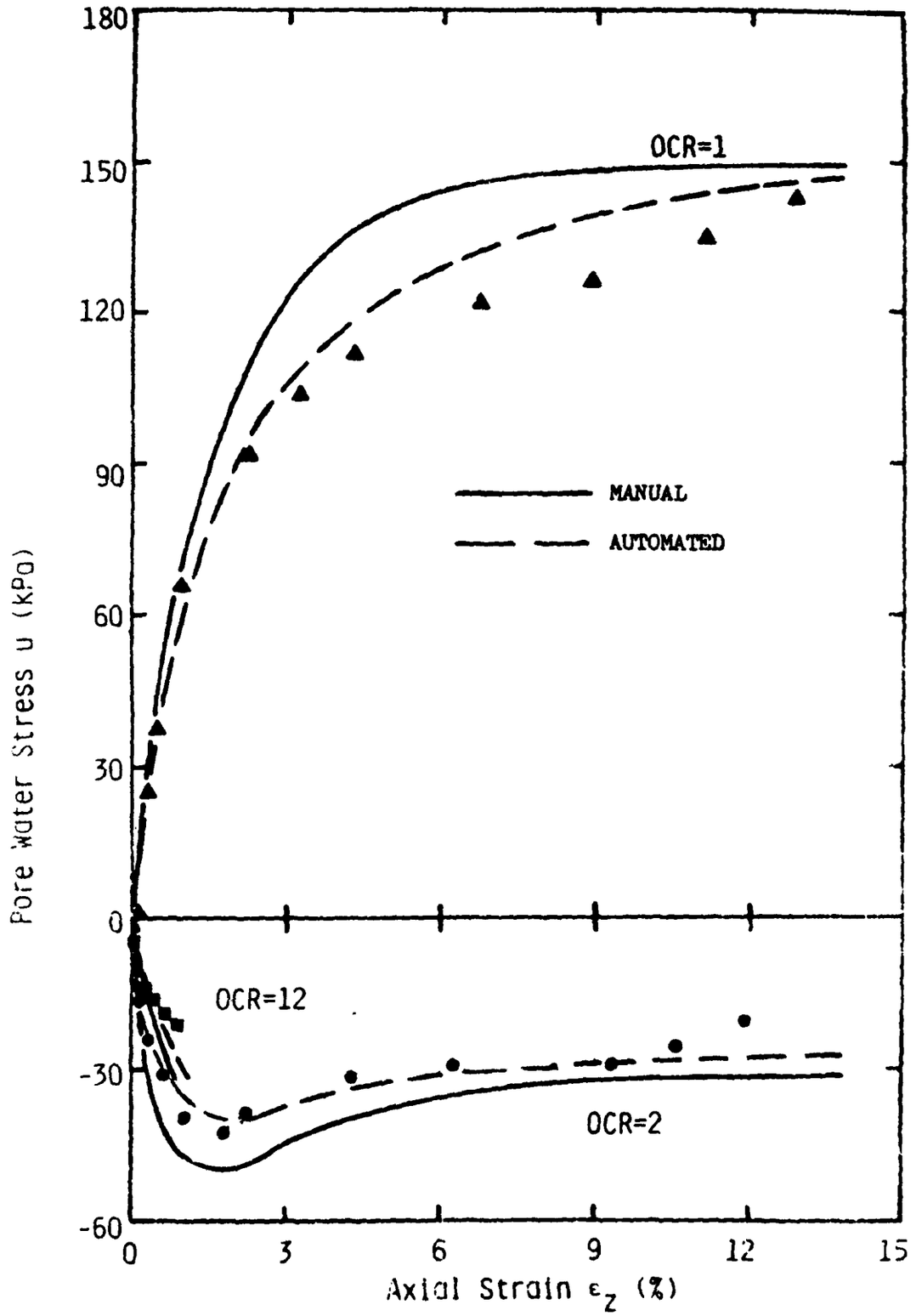


Figure 6 : Calibration Curves for u vs. ε_z Relation in Triaxial Extension

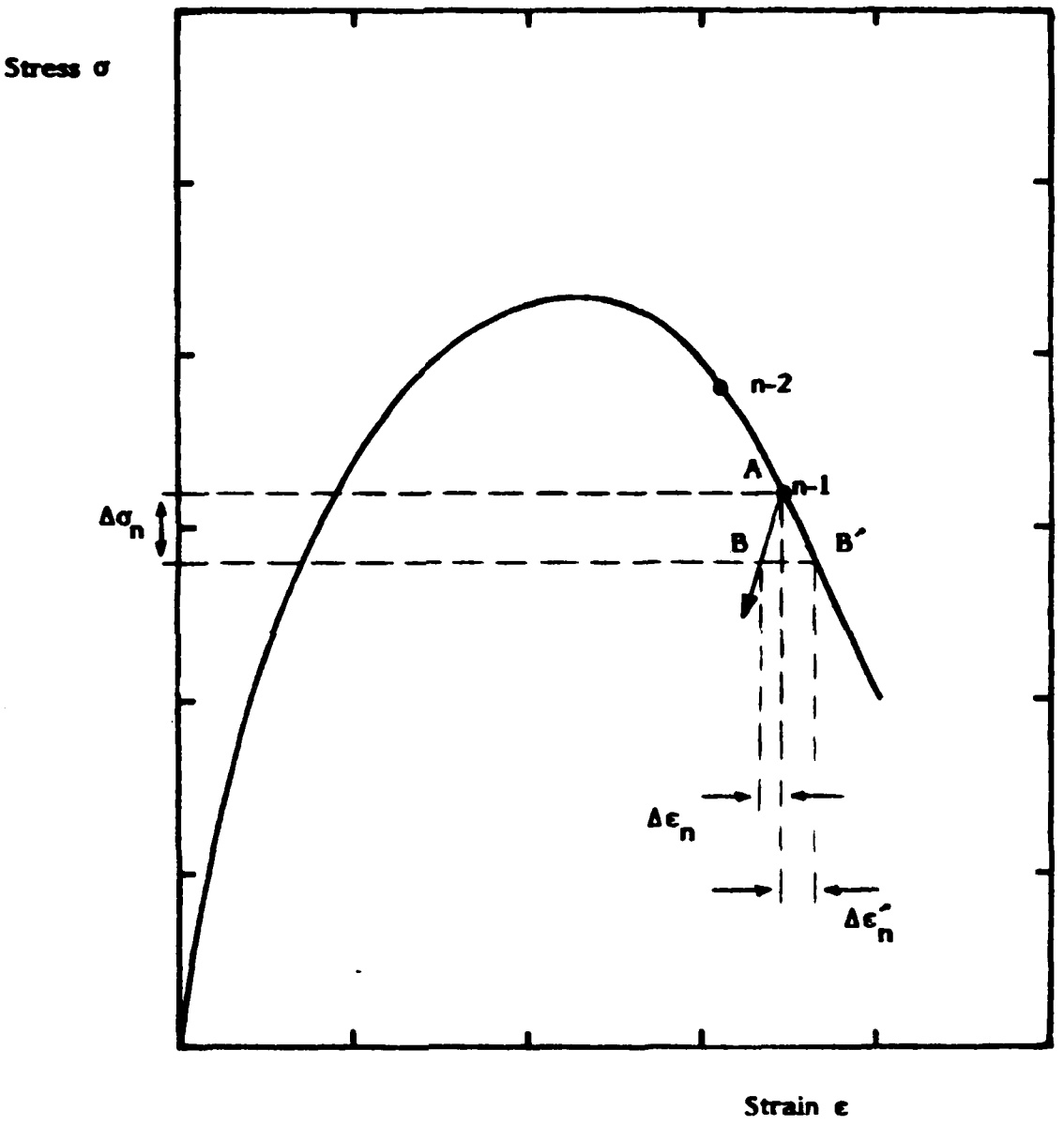
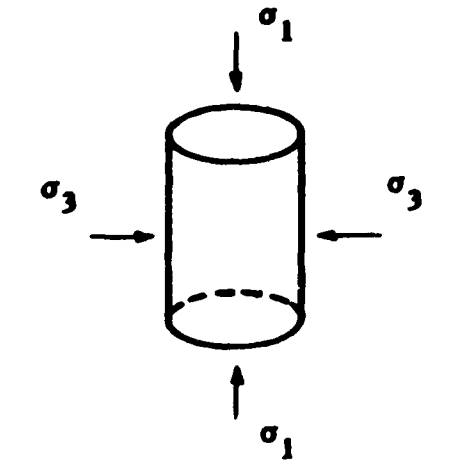
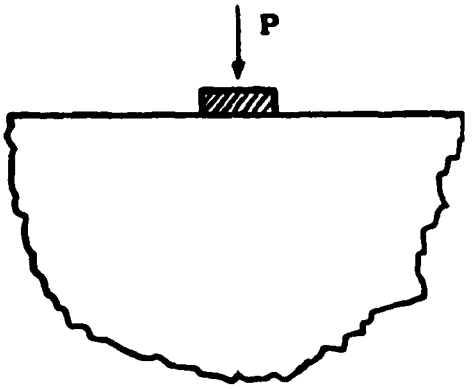


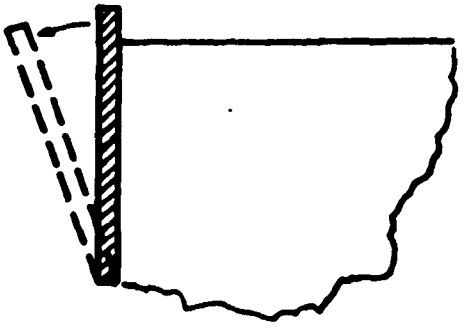
Figure 7: One-Dimensional Soil Response



Problem 1:
Triaxial Test

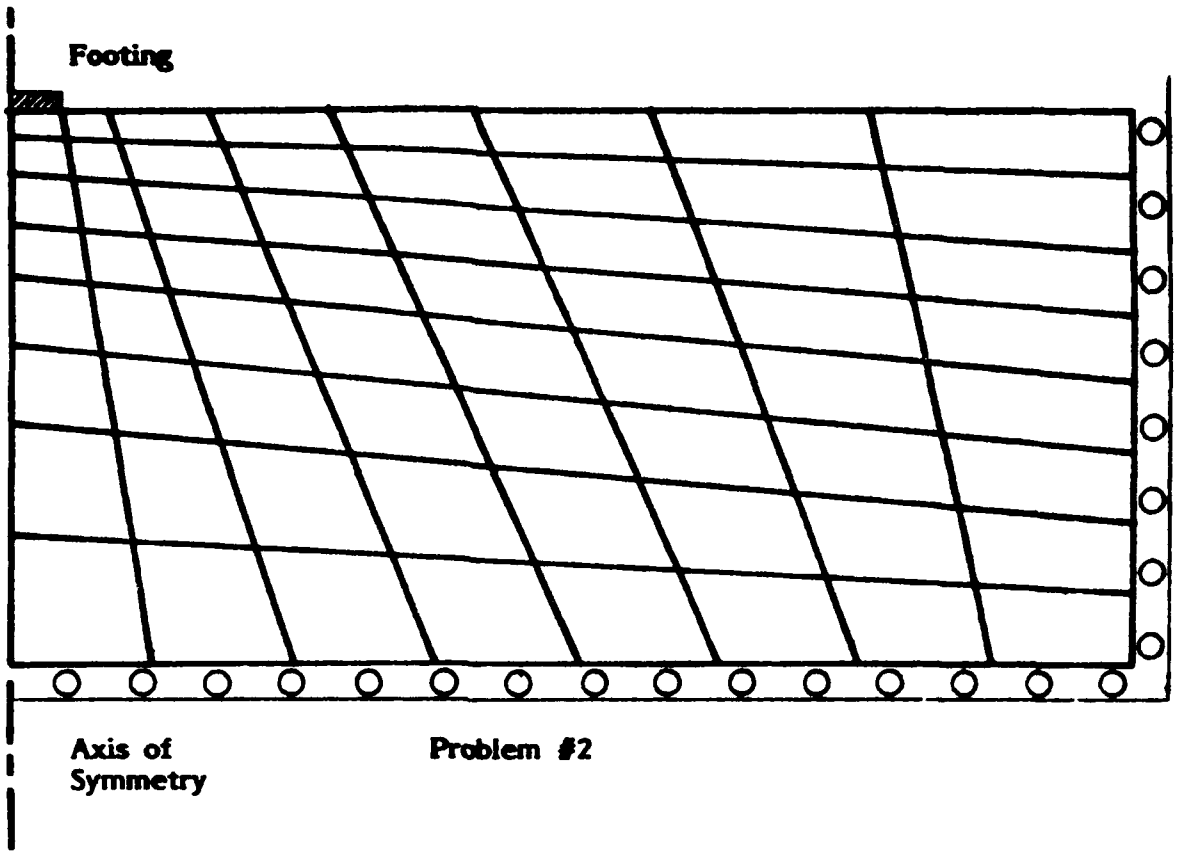


Problem 2:
Loaded Footing



Problem 3:
Rotation of A
Retaining Wall

Figure 8: Problem Configurations



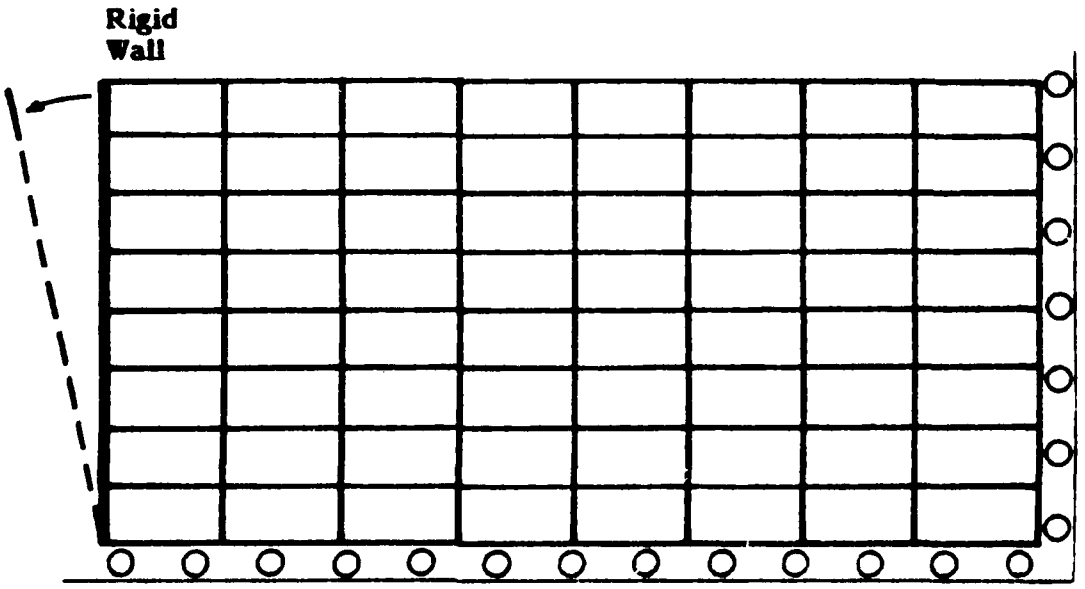


Figure 9: Finite Element Grids

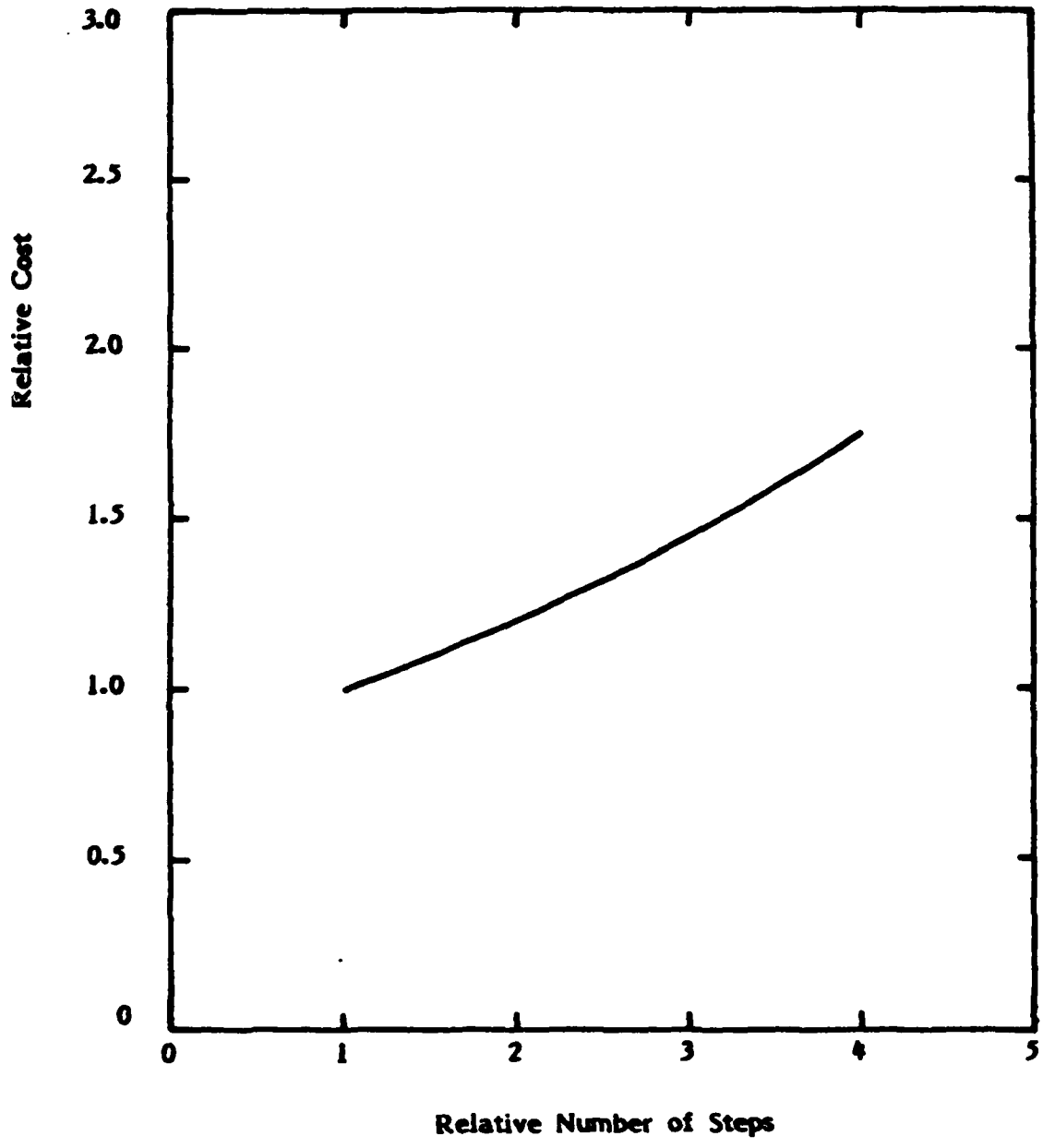


Figure 10: Computational Cost As A Function of Solution Step Size

APPENDIX A

NON-HOMOGENEOUS MODEL AND IN SITU TEST RESULTS

To date, the bounding surface model has been used primarily to predict the behavior of soils under homogeneous states of stress and strain. In these applications the model has been shown to be accurate and versatile. However, in order to fully verify its predictive capabilities and practical usefulness, it is necessary to incorporate the material model into a nonlinear finite element code and study its characteristics when applied to nonhomogeneous laboratory or in situ boundary-value problems.

The degree to which this final step in the model verification process can be completed is limited by the rather sparse amount of published experimental data. Ideally, homogeneous test results should be provided to permit the bounding surface model to be fully calibrated. Model test or field measurements must also be presented to serve as a measure of the formulation's predictive capabilities for non-homogeneous stress-strain conditions. Few published studies meet both requirements, although it is often possible to acquire all necessary data by consulting several articles by the same research group. Some of the reported experimental data that may be used in this final stage of the verification process is briefly discussed in the following paragraphs.

Ajaz and Parry (1976) describe the response of a compacted natural clay beam subjected to a series of laboratory bending tests. The testing apparatus was designed to permit a study to be made of the material's tensile strength and stress-strain response in uniform bending. The laboratory results are presented in the form of a moment-deflection curve for the center section, and by a series of strain contours corresponding to different applied bending moments. By assuming a particular set of equilibrium and strain-displacement relations, the authors are able to portray their results in terms of the soil's stress-strain response. Plane strain conditions are assumed. Unfortunately, none of the necessary soil properties are provided, and there is not

sufficient information to calibrate the bounding surface model. Additional aspects of the authors's research are presented in other published articles, and it may be possible to acquire the required calibration information by consulting these sources.

Baasubramian, Sivandran and Ho (1979) describe the results of three separate full-scale in situ slope stability tests involving Bangkok clay. One of these embankments (specifically, their "Embankment I") could be analyzed with a nonlinear, two-dimensional, plane-strain finite element program. The experimental results are presented in the form of force-deformation and force-pore water pressure histories for various locations at the test site. Although not all of the necessary material constants are provided, Bangkok clay has been extensively tested in the laboratory, and therefore a full complement of material properties could be readily acquired by consulting additional cited references.

Hanzawa (1979) describes a combined laboratory and in situ testing program conducted with a natural clay. The study is concerned strictly with ultimate strength, and an attempt is made to identify the effects of such quantities as consolidation history, strain rate and aging on shear strength. No complete stress-strain response is provided, and no field or model tests are reported.

Desai et al (1981) describe the results of a series of homogeneous and model tests on an artificial soil made of oil, sand and clay. Conventional triaxial compression and extension test results are provided, together with values of the traditional critical state parameters λ, κ and M; the bounding surface model could be readily calibrated for normally consolidated conditions. Two separate bearing capacity tests were conducted with scale models, and the results are report in the form of force-displacement curves. Both tests could be analyzed with a nonlinear, two-dimensional, plane-strain finite element program. Additional experimental results may also be available in other published and referenced articles.

Andersen and Stenhamar (1982) reported the results of three in situ static plate loading tests on heavily overconsolidated Haga clay. The OCR profile of the natural deposit is given and the experimental results are presented in the form of force-displacement and force - pore water pressure histories. Either plane strain or axisymmetric conditions are assumed, and the tests could be analyzed with a nonlinear, two-dimensional finite element code. The values of the critical state parameters λ, κ and M are not provided, but could be obtained by consulting other references. Otherwise, there is enough homogeneous laboratory data to enable the Bounding Surface model to be calibrated for the heavily overconsolidated range and compressive stress states.

Radhakrishnan and Reese (1969) report the results of a laboratory model study in which they studied the response of homogeneous and two-layered clay masses beneath a loaded strip footing. The experimental observations are presented in the form of force-deformation histories at various locations beneath the loaded footing. No consolidation or drained compression data is provided, and the initial state of the material (e_o, p'_o σ'_{co}) is not specified. The index properties of the two natural clays are tabulated, and deviator stress-axial strain relations from unconsolidated, undrained, (UU) triaxial compression tests at three different confining pressures are presented from which some of the necessary model parameters could perhaps be found. Both model tests could be analyzed with a nonlinear, two-dimensional, plane-strain finite element formulation. Additional material properties may possibly be found in cited references.

APPENIX B

USER'S MANUAL FOR NTD

Nonlinear 2-D Stress Analysis Program (Four Node Element)

by

L. R. Herrmann

Department of Civil Engineering

University of California, Davis

August, 1982

PART I: INPUT:

The required input data is entered by means of the following sequence of cards:

A1. Title Card (18A4):

Any information that is to be printed as the title of the problem.

A2. Control Card (2I5):

The following card is required to define the desired analytical options:

Columns			
5	**MTYPE**	=	0 - plane stress analysis 1 - plane strain analysis 2 - axisymmetric analysis
10	**IHISBF**	=	History function number (corresponding to the history function specifications of section B1) for the body force terms

A3. Nonlinear Analysis Control Card (5I5, 2E10.3): The following card is required to specify the desired iteration options:

Columns			
1 - 5	**NONLIN**	=	1 - successive approximations 2 - tangent stiffness (Newton's method)
6 - 10	**ITMAX**	=	Maximum number of iterations permitted in any single solution increment
11 - 15	**NWAY**	=	0 - no local iteration 1 - with local iteration
16 - 20	**IRPET**	=	0 - reform every iteration K - reform every K-th iteration
21 - 25	**ITFAC**	=	0 - no acceleration 1 - constant acceleration factor = RELAX 2 - variable norm acceleration 3 - variable component acceleration
26 - 35		=	$\left\{\begin{matrix}- \\ \text{RELAX} \\ C_L\end{matrix}\right\}$ for ITFAC = $\left\{\begin{matrix}0 \\ 1 \\ 2,3\end{matrix}\right.$
36 - 45	**ERMAX**	=	Convergence criterion for the displacement vector (by default, ERMAX = 0.01)

B1. A card with a 1 punched in column 1 followed by:

History Function Descriptons: The following cards are required for each distinct function:

1st Card (1X, I4, I5):

Columns		
2-5	IH	= Function number
6 - 10	M	= Number of points needed to define the function

2nd Card(s) (8E10.3):

As many cards as needed to specify the M pairs of values (F_m, t_m). The initial card should contain the values F_1, t_1, F_2, t_2,..., F_4, t_4. Subsequent cards, if required (M > 4), should contain the values F_5, t_5,..., F_M, t_M.

B2. **A card with a 2 punched in column 1, followed by:**

Material Properties Array: The following information must be supplied for each distinct material:

1st Card (1X, I4, I5, 2E10.3):

Columns			
2 - 5	MNAT	=	Material number
10	ITYP	=	1 - isotropic linear-elastic 2 - anisotropic linear-elastic 3 - bounding surface plasticity model for cohesive soil
11 - 20	$F_{x(r)}$	=	magnitudes of the body force components
21 - 30	$F_{y(z)}$	=	

2nd Card (8E10.3):

Columns	ITYP = 1	ITYP = 2	ITYP = 3
1 - 10	E	D_{11}	λ
11 - 20	ν	D_{12}	κ
21 - 30		D_{13}	M_c
31 - 40		D_{14}	R_c
41 - 50		D_{22}	A_c
51 - 60		D_{23}	T
61 - 70		D_{24}	p_ℓ
71 - 80		D_{33}	p_o

3rd Card (8E10.3) — required only if ITYP > 1:

Columns	ITYP = 1	ITYP = 2	ITYP = 3
1 - 10		D_{34}	m_c
11 - 20		D_{44}	h_c
21 - 30			e_o
31 - 40			G or ν
41 - 50			Γ
51 - 60			p_a
61 - 70			$\eta = M_e/M_c$
71 - 80			$\mu = h_e/h_c$

4th Card (5E10.3) — required only if ITYP = 3:

Columns	ITYP = 1	ITYP = 2	ITYP = 3
1 - 10			s
11 - 20			$r = R_e/R_c$
21 - 30			$a = A_e/A_c$
31 - 40			$t = m_c/m_c$
41 - 50			c

B3. A card with a 3 punched in column 1, followed by:

Initial Stress Information: The following information must be supplied for each initial stress state:

1st Card (1X, I4, 6E10.3):

Columns			
2 - 5	ISNO	=	Stress state number
6 - 15	a_1	=	vertical stress distribution,† in the form $\sigma_v = a_1 + a_2 y$
16 - 25	a_2		
26 - 35	b_1	=	horizontal stress distribution,† in the form $\sigma_h = b_1 + b_2 y$
36 - 45	b_2		
46 - 55	c_1	=	pore water stress distribution,† in the form $u = c_1 + c_2 y$
56 - 65	c_2		

2nd Card(s) (8E10.3)

As many cards as needed to specify the M pairs of values (t_m, F_m). The initial card should contain the values t_1, F_1, t_2, F_2, . . ., t_4, F_4. Subsequent cards, if required (M > 4) should contain the values t_5, F_5, . . ., t_M, F_M.

† Assuming plane conditions. For axisymmetry, the vertical coordinate direction is represented by z instead of y, and the distributions are then of the form $\sigma_v = a_1 + a_2 z$, etc.

B4. **A card with a 4 punched in column 1, followed by:**

Node Point Array (1X, I4, 2E10.3, I5, 3E10.3): As many cards as are necessary to specify the locations of all nodes in the system:

Columns				
2 - 5	N	=	Node point number	
6 - 15	XN	=	x - coordinate †	
16 - 25	YN	=	y - coordinate †	
26 - 30	INC	=	Numbering increment	quantities associated with the straight and curved line generation options
31 - 40	D	=	Spacing ratio	
41 - 50	XC	=	Coordinates of a point along the interior of the circular arc †	
51 - 60	YC			

† Assuming plane conditions. For axisymmetry, x → r and y → z.

B5. A card with a 5 punched in column 1, followed by:

Element Array (IX, I4, 8I5): As many cards as are necessary to specify all elements in the system:†

Columns			
2 - 5 6 - 10 11 - 15 16 - 20		=	The numbers of the four node points which describe the quadrilateral or triangular†† element (reading counter-clockwise around the element)
21 - 25	MN	=	Material number (corresponding to the material descriptions of section B2)
26 - 30	ISNO	=	Initial stress state number (corresponding to the stress state descriptions of section B3)
31 - 35	NMIS	=	Number of additional elements in the layer
36 - 40	INC	=	Numbering increment for elements within the layer
41 - 45	NMISP	=	Number of additional layers
46 - 50	INCP	=	Numbering increment for layers

quantities associated with the element generation option

† The order of these element cards need bear no relation to the actual location of the elements within the body.

†† For a triangular element the forth node number is set equal to the first.

B6. **A card with a 6 punched in column 1, followed by:**

Node Point Specification Array (1X, I4, 2 (I3, I2, E10.3), E10.3, 2I5, 2E10.3): As many cards as are necessary to specify the applied nodal displacements and loads:

Columns			
2 - 5	N	=	Node Point number
6 - 8	IH_1	=	History function number (corresponding to the history function specifications of section B1) for the 1-coordinate direction.
10	IF_1	=	$\left.\begin{matrix}0\\1\end{matrix}\right\}$ indicates that an applied $\left\{\begin{matrix}\text{force}\\\text{displacement}\end{matrix}\right.$ is specified in the 1-coordinate direction
11 - 20	V_1	=	Value of the $\left\{\begin{matrix}\text{force}\\\text{displacement}\end{matrix}\right.$ applied in the 1-coordinate direction
21 - 23	IH_2	=	History function number (corresponding to the history function specifications of section B1) for the 2-coordinate direction
25	IF_2	=	$\left.\begin{matrix}0\\1\end{matrix}\right\}$ indicates that an applied $\left\{\begin{matrix}\text{force}\\\text{displacement}\end{matrix}\right.$ is specified in the 2-coordinate direction
26 - 35	V_2	=	Value of the $\left\{\begin{matrix}\text{force}\\\text{displacement}\end{matrix}\right.$ applied in the 2-coordinate direction
36 - 45	θ	=	Angle (in degrees) between the x_1-axis and x(r)-axis
46 - 50	N'	=	Final node point in the sequence
51 - 55	INC	=	Numbering increment for node points within the sequence
56 - 65	P_N	=	Values of the pressures applied at points N and N′ respectively
66 - 75	$P_{N'}$		

} quantities associated with the boundary condition generation option

B7. **A card with a 7 punched in column 1, followed by:**

Solution History Segment Information (1X, I4, 2E10.3): One card for each history segment into which the incremental analysis is to be subdivided:†

Columns			
2 - 5	**NMIS**	=	Number of solution (or time) increments into which the history segment is to be subdivided
6 - 15	**TIME**	=	Solution time at the end of the history segment
16 - 25	**D**	=	Incrementing ratio defining the time-step lengths within the history segment (by default, D = 1.0)

B8. End Card (I1): A card with an 8 punched in column 1 to denote the end of the input data for given problem.

C1. **The above sequence of cards A1 → B8 should be repeated if additional analyses are desired.**

† note that the analysis begins at time $t_o = 0$)

PART II: EXPLANATORY NOTES REGARDING THE INPUT

Nonlinear Analysis Strategy (section A3):

The choice among the available solution strategies is selected by the specifications on this card. The reader is referred to Section 3.5 of the main body of the report for a description of the available options.

History Function Descriptions (section B1):

The histories of the body forces and applied node point displacements and loads are specified by means of "history functions". These history functions must belong to one of the following three classes:

i) IH < 0: specifies identical incremental values which are equal to the specified force/displacement. The incremental values are taken to be equal regardless of the relative lengths of the time history steps specified in section B7.

ii) IH = 0: specifies a step-function history at time t_1=0; that is, the specified force/displacement is applied entirely during the first solution increment, and no additional load/displacement is applied during the remaining solution increments.

iii) IH > 0: specifies the particular history function IH defined in section B1. The form of these functions is illustrated in Figure 1A and discussed below.

Consider the case of IH > 0. At time t_1=0 the function $F(t_1) = F_1$ need not necessarily be zero. For a step function load (at t_1=0 or at any other time t_m), the history segment must be described as a very steep ramp (that is, $t_m - t_{m-1} \simeq$ small but $\neq 0$) in section B1. The solution segment must be similarly defined in section B7. Within a particular history segment, linear interpolation is used to identify the ΔF which corresponds to the given time increment Δt. For solution times beyond the last specified point t_M, the final history segment is extended indefinitely. If a value V and a history function number IH > 1 is specified in Section B6 for some given external agent, then in the solution interval Δt an incremental value of the quantity equal to $V\Delta F$ is applied, where ΔF (Figure 1B) corresponds to history function IH.

Material Properties Array (Section B2):

For general linear, anisotropic, elastic material behavior the stress-strain relation is of the form:

$$[\sigma] = [D]\ [\epsilon]$$

For plane stress or plane strain conditions, the stress and strain vectors are defined by:

$$[\sigma]^T = [\sigma_x\ \sigma_y\ \sigma_z\ \tau_{xy}]^T$$

$$[\epsilon]^T = [\epsilon_x\ \epsilon_y\ \epsilon_z\ \gamma_{xy}]^T$$

while for axisymmetry (with respect to the z-axis), the vectors become:

$$[\sigma]^T = [\sigma_r\ \sigma_z\ \sigma_\theta\ \tau_{rz}]^T$$

$$[\epsilon]^T = [\epsilon_r\ \epsilon_z\ \epsilon_\theta\ \gamma_{rz}]^T$$

The 21 material properties required to define clay type materials (ITYP = 3) are defined in Herrmann et al (1980)†.

Initial Stress Information (Section B3):

In specifying the initial stress profiles, it is assumed that the coordinate system is aligned in the following as shown in Figure 2B.

†Herrmann, L. R., Dafalias, Y.F. and DeNatale, J.S. (1980), "Bounding Surface Plasticity for Soil Modelling," Department of Civil Engineering, University of California, Davis, Final Report to the Naval Construction Battalion Center, Port Hueneme, CA.

Node Point Array (Section B4):

The program incorporates two data generation routines to assist the user in defining the locations of the system's node points. The use of these options can, for example, enable one to describe the nodal geometry of an arbitrarily large grid with as few as five cards. Note that not all numbers between 1 and the maximum node number need correspond to actual nodes in the body. For example, the numbering scheme shown in Figure 3B is permissible, and the coordinates of the non-existent nodes 15 and 21 may or may not be specified. This feature facilitates the use of the node point and element generation options defined.

The straight line or circular arc coordinate generation option may be used whenever several sequential node points lie along a single straight line or circular arc. If such a situation exists, it is necessary only to enter the coordinates of the initial and final points of the sequence (denoted by N′ and N, respectively), and the values of INC and D. The constant INC represents the difference between any two successive node numbers in the sequence, and D defines the ratio of the distances between any two adjacent pairs of points.

If, for a node N, INC $\neq$ 0, intermediate node points are generated along a straight line (XC = YC = 0) or a circular arc (XC $\neq$ 0 and/or YC $\neq$ 0) between node N and the point described on the preceeding node specification card N′. That is, the coordinates of the points N′ + INC, N′ + 2*INC, . . ., N - INC are each automatically found. A circular arc is assumed to pass through the end points of the sequence N′ and N, and some additional intermediate point having coordinates (XC, YC). This intermediate point need not necessarily be a node.

The end points of the sequence may be entered in either order. For example, the segments illustrated in Figure 4B could be defined by specifying the nodes in either the order 7 → 22 (INC = 5) or the order 22 → 7 (INC = -5). The spacing of the intermediate points (nodes 12 and 17 in Figure 4B) is controlled by the spacing ratio D. The segments shown in Figure 4B could be generated by specifying either the order 7 → 22 and D = 2.0 or the order 22 → 7 and D = 0.5. A value of D = 1.0 would result in equally spaced nodes.

The interior node point generation option locates all interior nodes whose coordinates have not been established through the options cited above (that is, all points still left undefined after the section B4 input has been completed). The locations of these undefined interior nodes are computed by means of the "Laplacian - Isoparametric" grid generation scheme developed by Herrmann (1976).† In this scheme, the coordinates of an interior node are selected so as to represent a weighted average of the coordinates of the neighboring nodes.

Figure 5B illustrates two grids that have been prepared with the aid of the Laplacian - Isoparametric grid generation scheme. Grid 1 was developed by using the straight line generation routine to specify only the exterior (or boundary) nodes (and thus, only five cards were needed in section B4). Grid 2 was developed in an identical manner, except that the straight line generation option was also used to define the nodes lying along the line 3 → 21. Note that the exterior (or boundary) nodes must always be directly or indirectly specified in section B4.

Element Array (Section B5):

If the body can be divided into layers of elements, and if the material number MN and the initial stress state number ISNO is the same for several elements within a layer (or, perhaps, for several layers), the node numbers of these elements can be established by means of the element data generation option. To generate a sequence of elements within a single layer, node points are specified for the first element only, together with appropriate values of NMIS and INC (see section B5).

For example, the bottom row of elements in the grid of Figure 6A could be established by entering either the node numbers of element "A" and the values NMIS = 6 and INC = 4 or the node numbers of element "B" and the values NMIS = 6 and INC = -4. Similarly, the left-most column of elements could be established by entering the node numbers of element "A" and the values NMIS = 2 and INC = 1. Note, again, that the generated elements must be of the same material as the specified one.

† Hermann, L.R. (1976), "Laplacian - Isoparametric Grid Generation Scheme," Journal of the Engineering Mechanics Division ASCE, v. 102, no. EM5.

If several layers of elements are of the same material, it becomes possible to carry this option one step further. For example, the bottom two layers of elements in the grid of Figure 6B could be established by entering the node numbers of element "A" and the values:

NMIS = 6
INC = 4
NMISP = 1
INCP = 1

or, alternatively,

NMIS = 1
INC = 1
NMISP = 6
INCP = 1

Furthermore, if Grid 1 in Figure 5B represented a homogeneous body, the entire element array could be established by means of a single card containing, for example, the node numbers of element "A" and the values:

NMIS = 3
INC = 1
NMISP = 4
INCP = 5

Hence, under "ideal" conditions, the element array for a homogeneous body could be defined with only a single card in section B5.

Node Point Specification Array (Section B6):

Boundary or interior node point displacement and load specifications may be given in terms of either the x-y coordinate system (when $\theta = 0$ in section B6) or a local x_1-x_2 coordinate system (when $\theta \neq 0$), as shown in Figure 7B. If $\theta = 0$, the subscripts 1 and 2 in section B6 refer to x(r) and y(z) (and thus $IF_1 = IF_x$, etc.) and if $\theta \neq 0$ they represent x_1 and x_2 (and thus $IF_1 = IF_{x_1}$, etc.).

For each of the two coordinate directions, one may specify either a displacement (IF = 1) or a load (IF = 0) by setting V equal to the applied quantity. Specified displacements and loads are considered to be positive when they have the same sense as the positive coordinate directions. If a node is neither constrained nor subject to an applied load it need not -- and, for economy, should not -- be included in the node point specification array (section B6).

A uniform or linearly varying pressure may be specified along a straight or curved boundary (or an interior line) by means of the node point specification generation option. To use this option, the quantities IF_1, IF_2, V_1, V_2 and θ in section B6 are set to zero (or left blank), and the appropriate values of N, N', P_N and $P_{N'}$ are entered. For example, to specify the boundary loading shown in Figure 7, the user would enter, on a single input card, the values:

N'	=	11
N	=	2
INC	=	-3
P_N	=	100.0
$P_{N'}$	=	50.0

Note that the points N and N' must be specified in a counter-clockwise order if they lie on an exterior boundary and in a clockwise order if they lie along an interior boundary (or "hole"). Note also that the pressure specification cards must preceed all other node point specifications in section B6.

General Comments:

1. It is the responsibility of the user to maintain consistent units. The units used to describe the material properties (section B2) must be consistent with those used to describe the initial states of stress (section B3), the geometry of the body (section B4), and the applied loading (section B6). The solution will be expressed in terms of units which are consistent with those of the input specifications.

2. Because the bandwidth of the simultaneous equations is determined by the numbering of the nodes, an optimal node numbering scheme is required to minimize the computational costs of a given finite element analysis. The bandwidth resulting from a given numbering scheme may be computed in the following manner:

i) denote the bandwidth span for any two nodes of a given triangular (three node) or quadrilateral (four node) element as N_i, where N_i is equal to 2 plus twice the absolute difference in node numbers.

ii) denote the maximum value of N_i for a given element j as NE_j.

iii) considering all elements in the system, denote the maximum value of NE_j as NE_{max}.

Since NE_{max} represents the bandwidth of the simultaneous equations, in numbering the nodes it is this quantity that should be minimized.

3. As the program is now dimensioned, the value of NE_{max} must not exceed 64, the maximum node number (NPT) must not exceed 900, the number of elements (NELEM) must not exceed 841, the number of node point specifications (NBPTC) must not exceed 120, and the number of different materials (MNAT) must not exceed 5. When changing the dimensions of the program three separate areas must be considered:

   i) the COMMON blocks;

   ii) the values of KK = N_1 + 1 and LONG at the beginning of the code; and,

   iii) the dimension checks at the end of subroutine PREP.

The arrays used in the program which must be adjusted to accommodate larger analyses are related to problem size in the following manner:

i) $X(N_1)$, $Y(N_1)$ $NQ(N_1+1)$, $DISPLT(2N_1)$, $SL(2N_1)$, $SLP(2N_1)$, $SLPP(2N_1)$

ii) $NOD(N_2,4)$, $MNO(N_2)$

iii) $NODB(N_3)$, $BIV(N_3, 3)$

iv) Q7(LONG)

v) $PROP(N_4, 21)$, $FXA(N_4)$, $FYA(N_4)$, $ITYPA(N_4)$

where:

N_1 = maximum node number
N_2 = maximum number of elements
N_3 = maximum number of node point specifications
N_4 = NE_{max}
N_5 = maximum number of different materials

4. The size of the constant LONG (and the dimension of the Q7 array) must be large enough to satisfy the inequality:

$$LONG \geq f*(NE_{max})^2$$

where NE_{max} represents the bandwidth of the simultaneous equations. Although the value of LONG must satisfy the above relation for f = 1.0, it is recommended to specify LONG on the basis of f > 2.0.

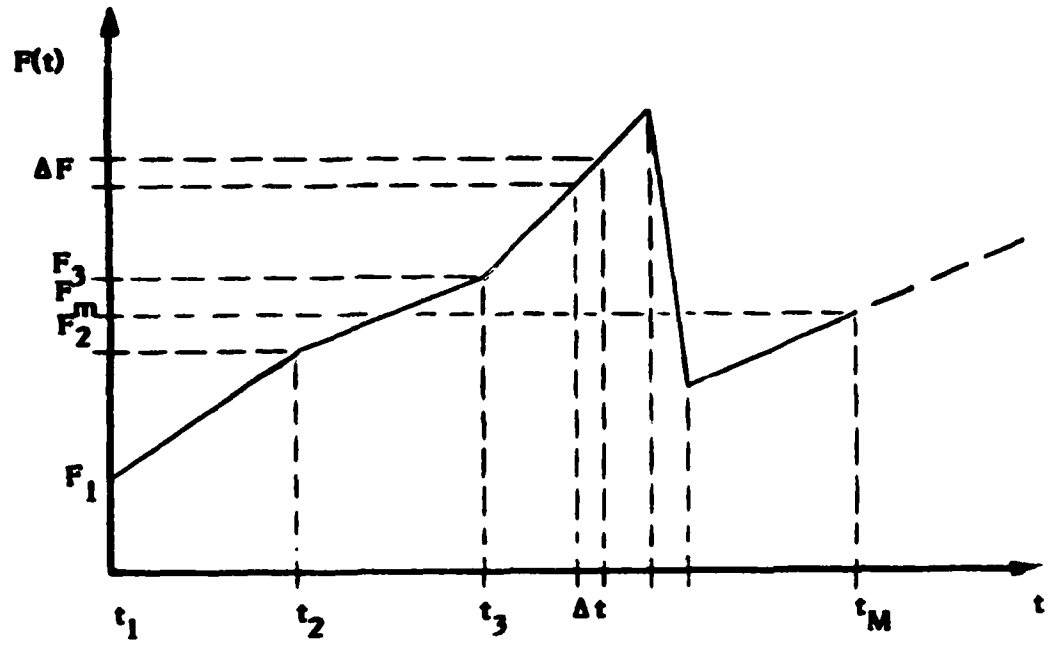


Figure 1B: Typical History Function

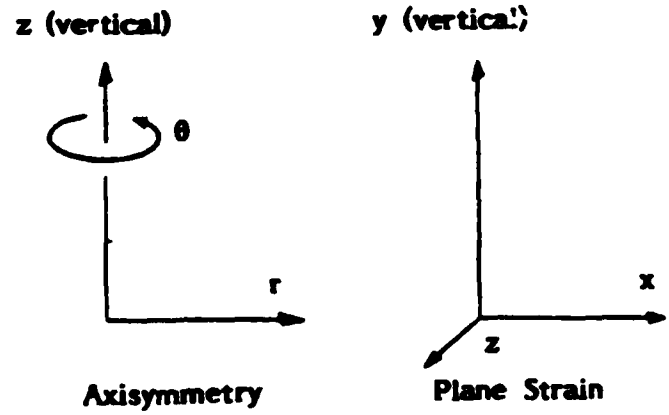


Figure 2B: Coordinate Alignments

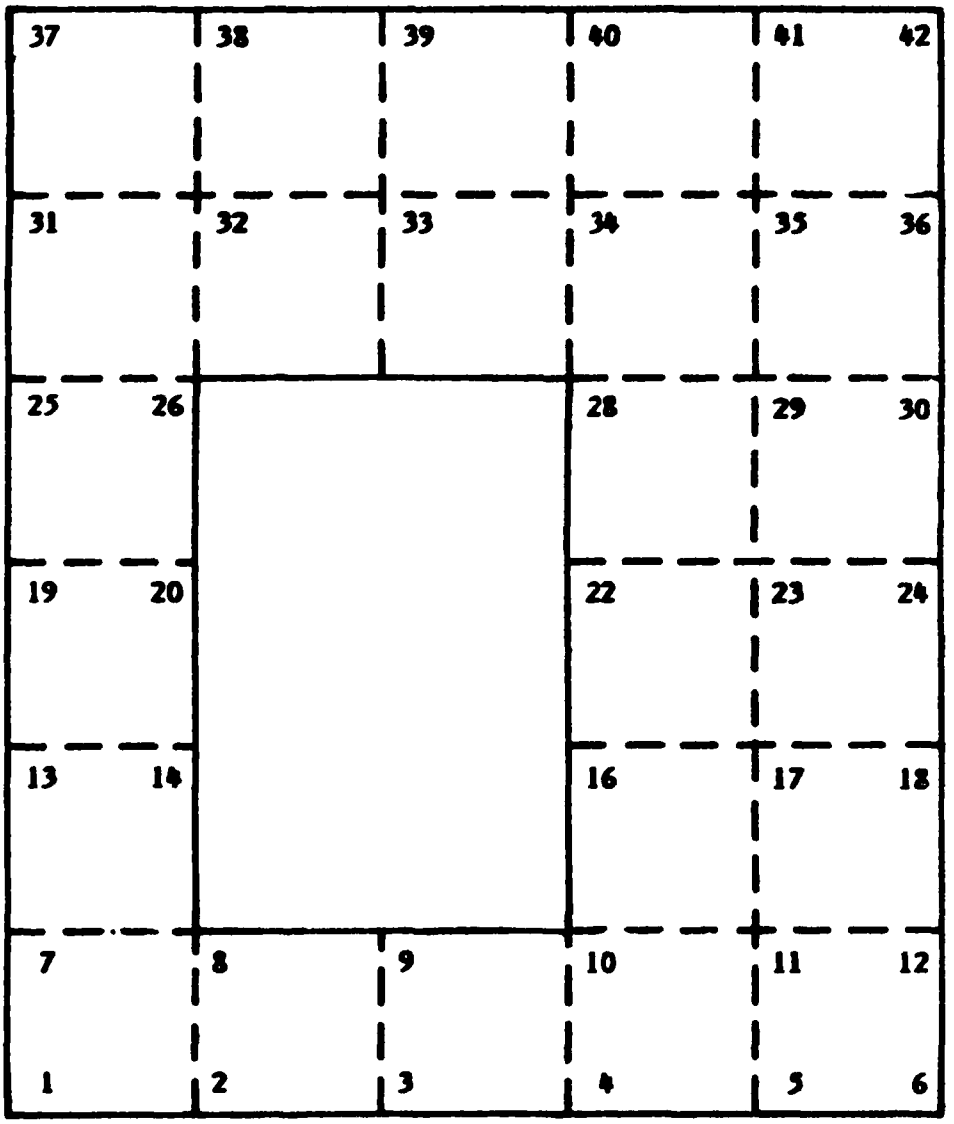


Figure 3B: Example of A Grid with Missing Node Numbers

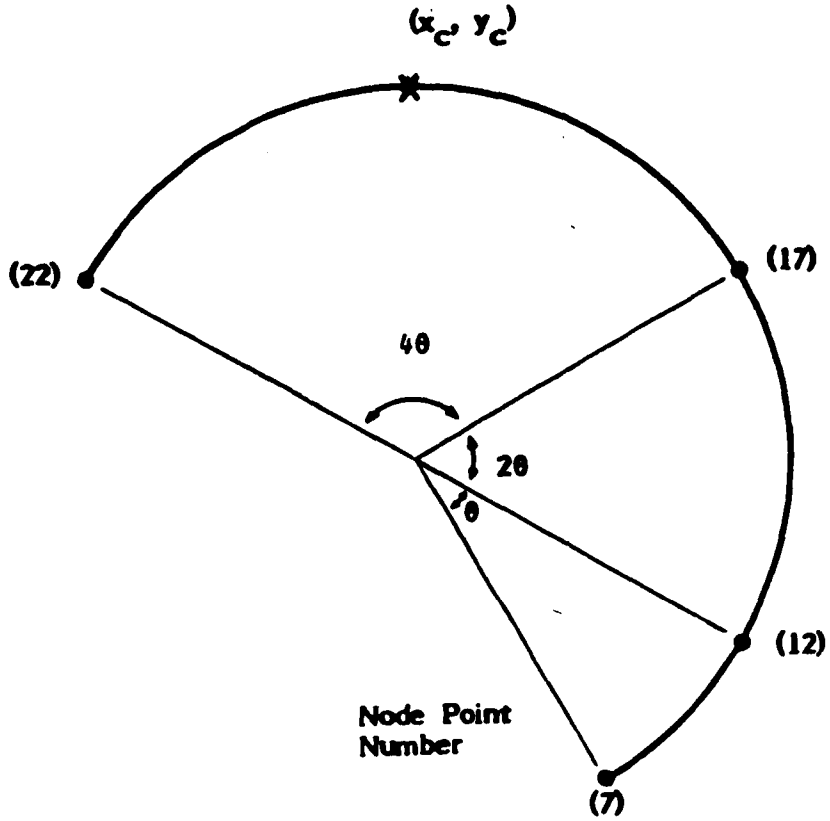


Figure 4Ba: Node Points Lying On A Circular Arc

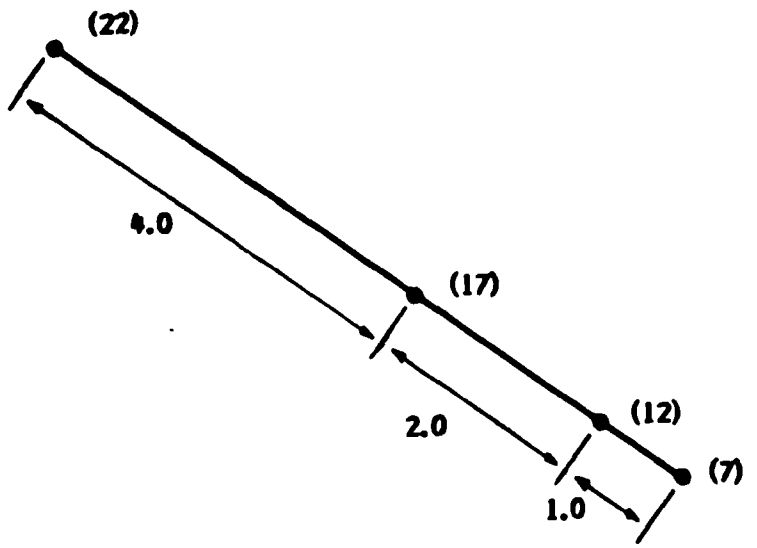
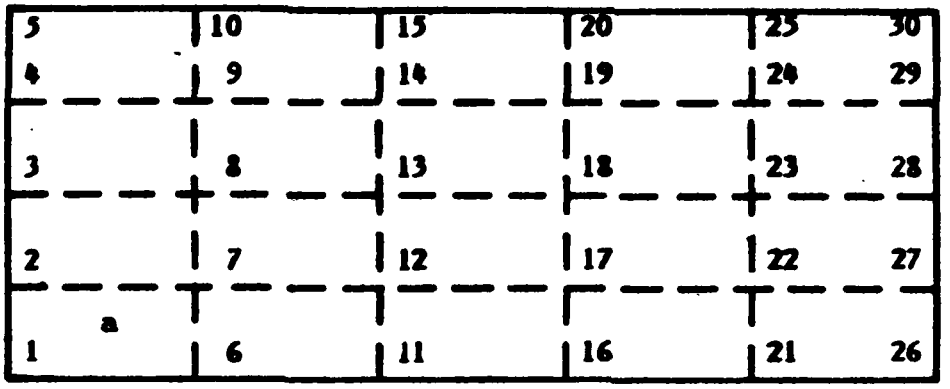
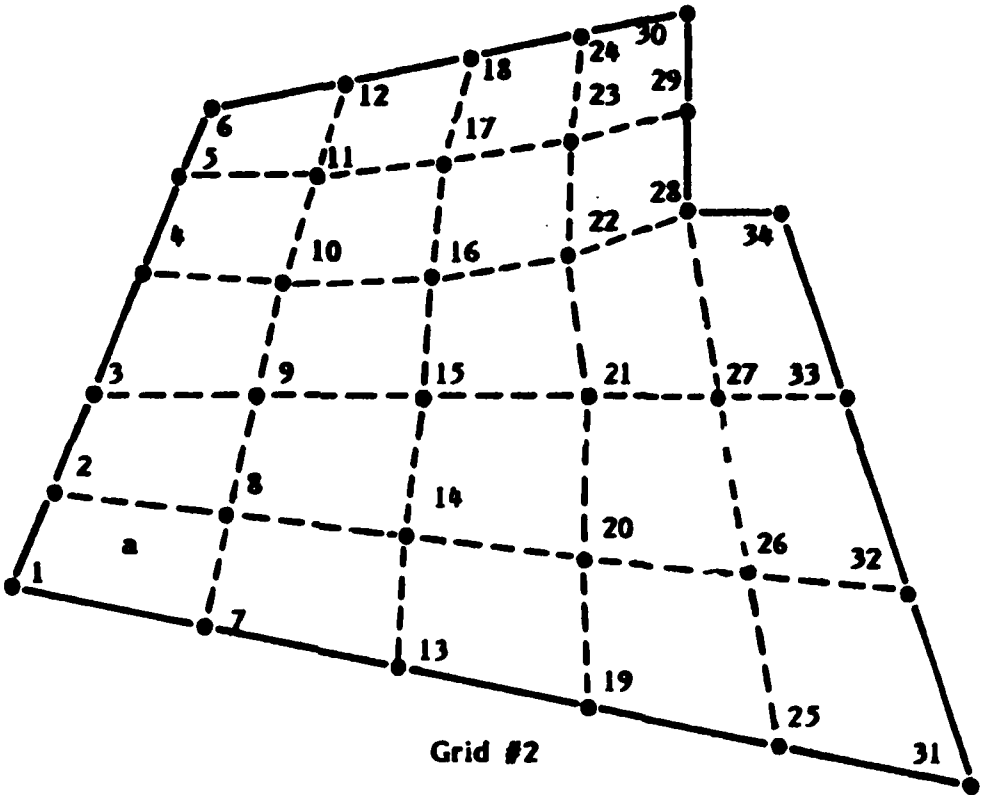


Figure 4Bb: Node Points Lying on A Straight Line



Grid #1



Grid #2

Figure 5B: Grids Prepared With The Aid Of The Generation Options

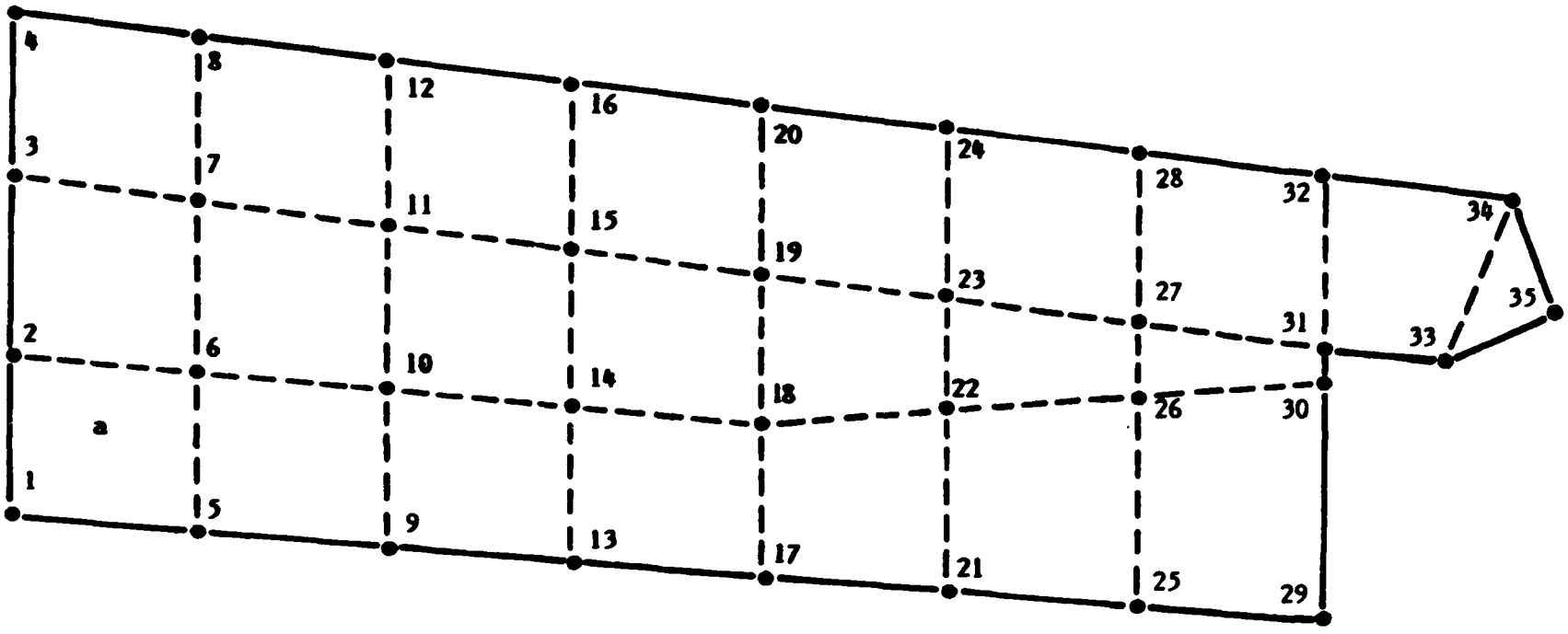


Figure 6B: Representation By Quadrilateral Elements

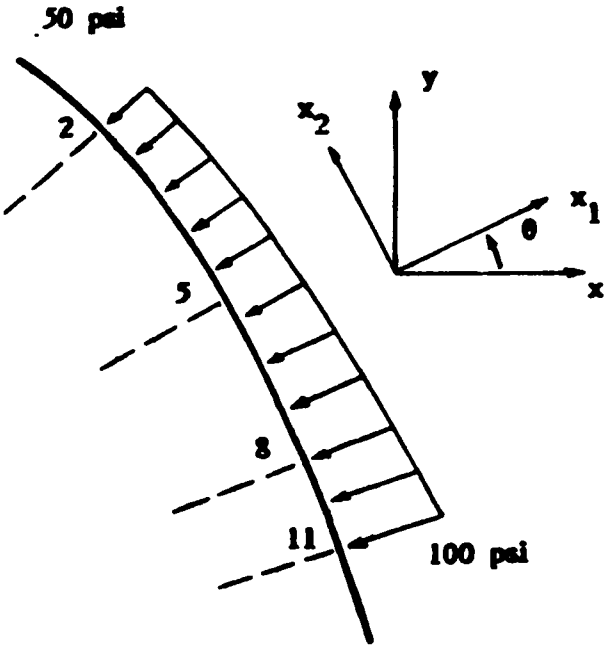


Figure 7Bc Pressurized Boundary

LISTING

```
$RESET FREE
FILE  1(KIND=DISK, MAXRECSIZE=8200, BLOCKSIZE=1000, AREAS=20, AREASIZE=1)
FILE  2(KIND=DISK, MAXRECSIZE=150, BLOCKSIZE=150, AREAS=210, AREASIZE=1)
FILE  5(KIND=READER,FILETYPE=7)
FILE  6(KIND=PRINTER)
C
C         TWO-DIMENSIONAL NONLINEAR FINITE ELEMENT ANALYSIS
C
      COMMON Q7(8200)
      COMMON/BLK0/ NPT,NELEM,NBPTC,NTYPE,IHISBF,ITMAX,RELAX,ERMAX,
     *             VITFAC,MODNEW,ITFAC,NONLIN,NWAY,IREPET,ITNO,NSTP
     *             ,ITIM,TIMF,TIMB,XKP,XXX
      COMMON/BLK2/ X(900),Y(900),NQ(901),DISPLT(1800)
      COMMON/BLK3/ MNO(841),NOD(841,4)
      COMMON/BLK4/ NODB(120),BIV(120,3)
      COMMON/BLK5/ SL(1800),SLP(1800),SLPP(1800)
      COMMON/BLK6/ ROA(4),SCA(4),ETA(4)
      COMMON/CNT/ ICNT1,ICNT2,ICNT3
      DATA ROA,SCA,ETA /1.0,-1.0,1.0,-1.0,1.0,-1.0,-1.0,1.0,
     *                  1.0,1.0,-1.0,-1.0/
C
C         FORMAT STATEMENTS
C
  804 FORMAT(I1,I4,2E10.3)
  900 FORMAT(1X,I3,1X,'ITERATIONS WERE PERFORMED AT TIME',F8.3,
     *       2X,'THE ERROR AT THE END OF THE PROCESS WAS',F7.4)
  901 FORMAT(/,1X,'***** THE BANDWIDTH OF',I7,
     *       2X,'IS TOO LARGE FOR THE DIMENSION OF --LONG-- *****')
  902 FORMAT(1X,'*****CONVERGENCE DID NOT OCCUR*****')
  903 FORMAT(1H0)
C
C        THE SIZE OF THE EQUATION BLOCK IS SET
C
      LONG=8200
      XXX=1.0E+20
      KK=232
C
C     INPUT DATA IS READ
C
      DO 90 I=1,KK
   90 NQ(I)=0
C**********NOTE**********NOTE**************
      ICNT1=0
      ICNT2=0
      ICNT3=0
      SLNRM1.0.0
  100 IX=1
C**********NOTE**********NOTE*****
      WRITE(6,9000)ICNT1,ICNT2,ICNT3
 9000 FORMAT(1X,'NO TOTAL SOL=',I3,1X,'NO PARTIAL SOL=',I3,1X,'NO CALL
     1 CLAY=',I10)
      CALL PREP(IX,Q7(1))
C**********NOTE**********NOTE**************
      ICNT1=0
```

```
      ICNT2=0
      ICNT3=0
      IF(IX  .EQ. 1) GO TO 110
  105 READ(5,804)NSEC
      IF(NSEC .NE. 0)GO TO 100
      GO TO 105
C
C     THE EQUATION POSITIONING MATRIX FOR THE SYSTEM MATRIX IS FORMED
C
  110 N=2
      DO 160 I=1,NPT
  160 NQ(I+1)=0
      DO 198 M=1,NELEM
      DO 198 J=1,4
      I=NOD(M,J)+1
  198 NQ(I)=N
      NQ(1)=1
      DO 200 I=1,NPT
  200 NQ(I+1)=NQ(I)+NQ(I+1)
C
C       THE BANDWIDTH  -NCOL-  IS COMPUTED
C
      NCOL=0
      DO 218 I=1,NELEM
      DO 202 J=1,4
      JJ=NOD(I,J)
      DO 202 K=1,4
      KK=NOD(I,K) - 1
      IV=NQ(KK) - NQ(JJ)
      IF (NCOL .LT. IV)  NCOL=IV
  202 CONTINUE
  218 CONTINUE
C
C       COMPUTE MATRIX SPECIFICATIONS
C
      NROW=NQ(NPT + 1) - 1
      L2=NCOL-1
      L1=(LONG - L2*NCOL)/NCOL
      IF(L1 .GT. 0)  GO TO 220
      WRITE(6,901) NCOL
      GO TO 105
  220 IDISK=(NROW-1)/L1+1
      IF(NROW .GT. LONG/NCOL)  GO TO 222
      IDISK=0
      L1=NROW
      L2=0
  222 LT=L1 + L2
C
C     INITIALIZATION
C
      DO 225 I=1,NROW
      DISPLT(I)=0.0
      SLP(I)=0.0
  225 SL(I)=0.0
```

```
      ITIM=0
      TIMF=0.0
C
C        MARCHING TIME
C
  240 READ(5,804) NSEC,NMIS,TIMFS,D
      IF(NSEC .NE. 0)  GO TO 100
      IF(D .EQ. 0.0)  D=1.0
      DU2=NMIS
      DU1=1.0/DU2
      IF(D .EQ. 1.0)  GO  TO 244
      DU1=(1.0 - D)/(1.0 - D**NMIS)
  244 DT=(TIMFS - TIMF)*DU1
      DO 760 NSTPS=1,NMIS
      NSTP=NSTPS
      WRITE(6,903)
C
C     CHANGE DISPLACEMENT ESTIMATE AT BEGINING OF NEW SOLUTION
C     SEGMENT IN CASE OF UNSTABLE BEHAVIOR
C
      IF(NSTPS .GT. 1)GO TO 255
      DO 250 I=1,NROW
  250 SL(I)=-.01*SL(I)
  255 ITIM=ITIM + 1
      TIMB=TIMF
      TIMF=TIMF + DT
      DT=DT*D
C
C        ITERATION
C
      ITNO=-1
  260 ITNO=ITNO + 1
C
C     DECIDE IF SHOULD UPDATE STIFFNESS OR NOT
C
      MODNEW=1
      I=(ITNO-1)/IREPET
      IF(NSTPS .EQ. 1  .AND.  ITNO .EQ. 0)MODNEW=0
      IF(ITNO .EQ. I*IREPET + 1)MODNEW=0
C
C     INITIALIZE SYSTEM MATRIX
C
      DO 270 I=1,NROW
      SLPP(I)=SLP(I)
      SLP(I)=SL(I)
  270 SL(I)=0.0
      IF(MODNEW .EQ. 1)  GO TO 300
      LL=NCOL*LT
      DO 280 I=1,LL
  280 Q(I)=0.0
  300 CONTINUE
C**********NOTE**********NOTE*********
      IF(MODNEW .EQ. 0)ICNT1=ICNT1+1
      IF(MODNEW .EQ. 1)ICNT2=ICNT2+1
```

```
      CALL SOLVE (L1,L2,LT,NROW,NCOL,IDISK,Q7,Q7)
C
C       BOUNDARY POINTS TRANSFORMED TO X-Y COORDINATE SYSTEM
C
      DO 570 K=1,NBPTC
      KK=IABS(NODB(K))
      NODB(K)=KK
      ANG=BIV(K,3)
      IF(ANG .EQ. 0.0)  GO TO 570
      K1=KK/1000000
      JJ=NQ(K1)
      IF(JJ .EQ. NQ(K1+1))  GO TO 570
      CT=COS(ANG)
      SA=SIN(ANG)
      D1=SL(JJ)
      D2=SL(JJ-1)
      SL(JJ)=D1*CT-D2*SA
      SL(JJ+1)=D1*SA+D2*CT
  570 CONTINUE
C
C         NEED FOR ADDITIONAL ITERATION IS CHECKED
C
      ERNRM=0.0
      SOLNRM=0.0
      ITSTOP=0
      DO 580 I=1,NROW
C*********NOTE**********NOTE***************
      IF(NONLIN .EQ. 2)SL(I)=SL(I)+SLP(I)
      ERNRM=ABS(SL(I) - SLP(I)) + ERNRM
      SOLNRM=SOLNRM + ABS(SL(I))
  580 CONTINUE
C************NOTE***********NOTE********
      IF(ITFAC .NE. 2)GO TO 584
      RELAX=1.0
      IF(ITNO .EQ. 0  .OR.  (ITNO/2)*2 .NE. ITNO)GO TO 582
      CALL ACCEL(SLNRM2,SLNRM1,SOLNRM,RELAX,VITFAC)
  582 SLNRM2=SLNRM1
      SLNRM1=SOLNRM
  584 IF(SOLNRM .LE. 0.0)  SOLNRM=1.0E-20
      ERNRM=ERNRM/SOLNRM
      IF(ITNO .LT. ITMAX)  GO TO 590
      WRITE(6,902)
      GO TO 600
  590 IF(ERNRM .GT. ERMAX)  GO TO 650
  600 WRITE(6,900) ITNO,TIMF,ERNRM
      ITSTOP=1
      GO TO 700
  650 DO 670 I=1,NROW
C**********NOTE***********NOTE***********
      IF(ITFAC .NE. 3)GO TO 670
      RELAX=1.0
      IF(ITNO .EQ. 0  .OR.  (ITNO/2)*2 .NE. ITNO)GO TO 670
      DU=SL(I)
      DU1=SLP(I)
```

```
      DU2=SLPP(I)
      CALL ACCEL(DU2,DU1,DU,RELAX,VITFAC)
  670 SL(I)=SLP(I) + RELAX*(SL(I) - SLP(I))
C
C       STRESS AND STRAINS COMPUTED
C
  700 CALL STRESS (ITSTOP)
      IF(ITSTOP .EQ. 0)  GO TO 260
  760 CONTINUE
      GO TO 240
      END
C ***********************************************************************
      SUBROUTINE PREP (IX,SL)
C
C         THIS SUBROUTINE SETS UP THE DESCRIPTION OF THE PROELEM
C
      COMMON/BLK0/ NPT,NELEM,NBPTC,MTYPE,IHISBF,ITMAX,RELAX,ERMAX,
     *             VITFAC,MODNEW,ITFAC,NONLIN,NWAY,IREPET,ITNO,NSTP
     *             ,ITJM,TIMF,TIMB,XKP,XXX
      COMMON/BLK2/ X(900),Y(900),NQ(901),DISPLT(1800)
      COMMON/BLK3/ MNO(841),NOD(841,4)
      COMMON/BLK4/ NODB(120),BIV(120,3)
      COMMON/BLK7/ FUN(10,3),FUNT(10,3),NPTS(3,3)
      DIMENSION ANI(12),AN(3),TITLE(20),NODS(4),IS(8),HN(2),
     *          IIFLG(2),BIVD(2),IH(2),SCOEF(8,6),SL(1),SCOEFS(6)
C
C          FORMAT STATEMENTS
C
  800 FORMAT(5I5,2E10.3)
  801 FORMAT(8E10.3)
  802 FORMAT(I1,I4,9I5)
  803 FORMAT(I1,I4,6E10.3)
  805 FORMAT (20A4)
  806 FORMAT(I1,I4,2(I3,I2,E10.3),E10.3,2I5,2E10.3)
  808 FORMAT(I1,I4,2E10.3,I5,3E10.3)
  900 FORMAT (1H0 8X 20A4 ///)
  901 FORMAT(30X,'PLANE STRESS ANALYSIS',///)
  902 FORMAT(1H0,//////,35X,'***** ELEMENT INFORMATION *****',///,
     1         1X,'ELEMENT',9X,'ELEMENT',15X,'ELEMENT',
     2         12X,'MATERIAL',5X,'INITIAL STRESS STATE',/,
     3         1X,'NUMBER',10X,'CENTER',13X,'NODE POINTS',11X,
     4         'NUMBER',4X,'SIG-V', 6X,'SIG-H',8X,'U')
  903 FORMAT(30X, PLANE STRAIN ANALYSIS',///)
  904 FORMAT(2X,'SUCESSIVE SUBSTITUTION USED FOR NONLINEAR ANALYSIS')
  905 FORMAT(2X,'TANGENT STIFF METHOD USED FOR NONLINEAR ANALYSIS')
  906 FORMAT(I7,6X,2(A6,1PE10.2,A6,I3,8X),3X,A6,0PE10.3,/)
  907 FORMAT(30X,'AXISYMMETRIC ANALYSIS',///)
  908 FORMAT(1H0,9X,'*****GEOMETRY*****',/,'ONODAL POINT',6X,
     *       X-Y OR R-Z COORDINATES',/)
  909 FORMAT(I11,5X,1P2E10.2)
  910 FORMAT(//,15X,'HISTORY FUNCTION NO.',I3,/,17X,'VALUE',12X,'TIME')
  911 FORMAT           (//,1X,'ERROR-TOO MANY MATERIALS')
  912 FORMAT           (//,1X,'ERROR-TOO MANY ELEMENTS')
  913 FORMAT           (//,1X,'ERROR-TOO MANY NODE POINTS')
```

```
  914 FORMAT(//,1X,'ERROR-TOO MANY NODE POINT SPECIFICATIONS')
  915 FORMAT(1H0,//////,10X,'*****NODE POINT SPECIFICATIONS*****',///,
     *   4X,'NODE',/)
  916 FORMAT(2X,'THE ITERATION PROCESS HAS A LIMIT OF',I3,3X,
     *           'ITERATIONS PER INCREMENT',
     *       /,2X,'AND  A CONVERGENCE REQUIREMENT OF',F8.4)
  917 FORMAT(I5,5X,1P2E10.3,4I5,I12,0P3E12.3)
  918 FORMAT(2X,'***** DATA ERROR -- TOO MUCH HISTORY FUNCTION DATA',
     *         ' *****',//)
  919 FORMAT(12X,1PE12.3,5X,E12.3)
  920 FORMAT(//,' ERROR-DATA ERROR IN ELEMENT',I5)
  922 FORMAT(2X,'LOCAL ITERATION USED TO MODIFYTHE STRESS INCREMENT')
      DATA ANI,AN(3),HN /4HU-X=, 4HP-X=, 4HU-Y=, 4HP-Y=,
     *             4HU-R=,4HP-R=,4HU-Z=,4HP-Z=,
     *             4HU-1=, 4HP-1=, 4HU-2=, 4HP-2=,
     *             4HANG=, 4HIH1=, 4HIH2= /
  923 FORMAT(1X, STIFFNESS MATRIX UPDATED EVERY',I2,1X,'ITERATIONS',
     1         ' BEGINNING WITH SECOND')
  924 FORMAT(1X,'A CONSTANT ACCELERATION FACTOR=',F5.1,1X,'IS USED')
  925 FORMAT(1X,'A VARIABLE ACCELERATION FACTOR BASED ON THE'
     *     ' SOLUTION NORM AND WITH LIMITS OF',F5.2,1X,'AND',F5.2,
     *      1X,'IS USED')
  926 FORMAT(1X,'A VARIABLE ACCELERATION FACTOR BASED ON THE'
     *      INDIVIDUAL SOLUTION COMPONENTS AND WITH LIMITS OF',
     *      F5.2,1X,'AND',F5.2, 1X,'IS USED')
      DATA IS /1,2,3,4,1,2,3,4/
      DX=0.0
      DY=0.0
      PNF=0.0
      NMAT=0
      NPT=0
C
C       INPUT DATA IS READ
C
      READ(5,805,END=700) TITLE
      WRITE (6,900) TITLE
      READ(5,800) MTYPE,IHISBF
      READ(5,800)NONLIN,ITMAX,NWAY,IREPET,ITFAC,VITFAC,ERMAX
      IF(MTYPE .EQ. 0)  WRITE(6,901)
      IF(MTYPE .EQ. 1)  WRITE(6,903)
      IF(MTYPE .EQ. 2)  WRITE(6,907)
      XKP=0.0
      IF(MTYPE .EQ. 0)  XKP=1.0
      IF(ITMAX .LE. 0)  ITMAX=1
      IF(ERMAX .LE. 0.0)  ERMAX=0.01
      IF(RELAX .LE. 0.0)  RELAX=1.0
      IF(NONLIN .EQ. 1)WRITE(6,904)
      IF(NONLIN .EQ. 2)WRITE(6,905)
      IF(NWAY .EQ. 1)WRITE(6,922)
      IF(IREPET .EQ. 0)IREPET=1
      WRITE(6,923)IREPET
      WRITE(6,916) ITMAX,ERMAX
      IF(VITFAC .EQ. 0.0)VITFAC=1.0
      IF(ITFAC .EQ.0  .OR. VITFAC .EQ. 1.0)GO TO 8
```

```
      IF(ITFAC .GT. 1)GO TO 5
      RELAX=VITFAC
      WRITE(6,924)RELAX
      GO TO 8
    5 DU=1.0/VITFAC
      IF(ITFAC .EQ. 2)WRITE(6,925)VITFAC,DU
      IF(ITFAC .EQ. 3)WRITE(6,926)VITFAC,DU
    8 READ(5,802) NSEC
C
C         INPUT CONTROL UNIT
C
   10 GO TO (20,60,75,85,112,175,195),NSEC
C
C         TIME FUNCTIONS
C
   20 READ(5,802) NSEC,I,N
      IF(NSEC .NE. 0)  GO TO 10
      IF(I .LT. 4 .AND. N .LT. 11)  GO TO 22
      WRITE(6,918)
      GO TO 700
   22 NPTS(I,1)=0
      NPTS(I,2)=1
      NPTS(I,3)=N-1
      READ(5,801) ((FUN(J,I),FUNT(J,I)),J=1,N)
      WRITE(6,910) I
      WRITE(6,919) ((FUN(J,I),FUNT(J,I)),J=1,N)
      GO TO 20
C
C         MATERIAL PROPERTIES
C
   60 IN=1
      CALL PROPTY(IN,NZ,NMAT,NSEC)
      GO TO 10
C
C         INITIAL STRESS SPECIFICATIONS
C
   75 READ(5,803) NSEC,I,(SCOEFS(J),J=1,6)
      IF(NSEC .NE. 0)  GO TO 10
      DO 77 J=1,6
   77 SCOEF(I,J)=SCOEFS(J)
      GO TO 75
C
C         NODE POINT COORDINATES
C
   85 READ(5,806) NSEC,N,XP,YP,INCR,D,XC,YC
      IF(NSEC .NE. 0)  GO TO 10
      X(N)=XP
      Y(N)=YP
      NQ(N)=-2
      IF(INCR.EQ.0) GO TO 111
      IF(D.EQ.0.0) D=1.0
      NM=(N -NS)/INCR
      NMIS=IABS(NM)
      INCR=INCR*NM/NMIS
```

```
      DU2=NMIS
      DU1=1.0/DU2
      IF(D.EQ.1.0) GO TO 87
      DU1=(1.0-D)/(1.0-D**NMIS)
   87 IF(XC .EQ. 0.0 .AND. YC .EQ. 0.0) GO TO 95
C
C       GENERATE POINTS ON ARC
C
      C11=2.0*(XC-XS)
      C12=2.0*(YC-YS)
      C21=2.0*(XC-XP)
      C22=2.0*(YC-YP)
      DU=1.0/(C11*C22-C12*C21)
      B1=XC*XC-XS*XS+YC*YC-YS*YS
      B2=XC*XC-XP*XP+YC*YC-YP*YP
      X0=(C22*B1-C12*B2)*DU
      Y0=(C11*B2-C21*B1)*DU
      RC=SQRT((XC-X0)**2+(YC-Y0)**2)
      TH0=ATAN2((YS-Y0),(XS-X0))
      IF(TH0 .LT. 0.0)TH0=6.2831853+TH0
      DY=-(XP-X0)*SIN(TH0)+(YP-Y0)*COS(TH0)
      DX= (XP-X0)*COS(TH0)+(YP-Y0)*SIN(TH0)
      DTH=     ATAN2(DY,DX)
      ZC=(XC-XS)*(YP-YS)-(XP-XS)*(YC-YS)
      IF(ZC .GT. 0.0 .AND. DTH .LT. 0.0)DTH=6.2831853+DTH
      IF(ZC .LT. 0.0 .AND. DTH .GT. 0.0)DTH=-6.2831853+DTH
      DTH=DTH*DU1
      DO 90 I=2,NMIS
      TH0=TH0+DTH
      NS=NS+INCR
      NQ(NS)=-2
      X(NS)=X0+RC*COS(TH0)
      Y(NS)=Y0+RC*SIN(TH0)
   90 DTH=DTH*D
      GO TO 111
C
C         GENERATE POINTS ON STRAIGHT LINE
C
   95 DX=(XP  -XS)*DU1
      DY=(YP  -YS)*DU1
      DO 110 I=2,NMIS
      NS=NS+INCR
      XS=XS+DX
      YS=YS+DY
      NQ(NS)=-2
      X(NS)=XS
      Y(NS)=YS
      DX=DX*D
  110 DY=DY*D
  111 XS=XP
      YS=YP
      NS=N
      IF(NPT .LT. NS) NPT=NS
      GO TO 85
```

```
C
C        ELEMENT INFORMATION
C
  112 N=1
  113 READ(5,802) NSEC,(NODS(I),I=1,4),MNOR,ISIGN,NMISP,INCRP,NMIS,INCR
      IF(NSEC .NE. 0) GO TO 132
      DO 116 I=1,4
  116 NOD(N,I)=NODS(I)
      NS=N
      INCRS=0
      INCRZ=INCR
      NMISZ=NMISP
      MNO(N)=MNOR*100 + ISIGN
  120 DO 125 M=1,4
  125 NOD(N,M)=NOD(NS ,M)+INCRS
      MNO(N)=MNO(NS)
      N=N+1
      INCRS=INCRS+INCRP
      NMISP=NMISP-1
      IF(NMISP .GE. 0)GO TO 120
      NMISP=NMISZ
      INCRS=INCRZ
      INCRZ=INCRZ+INCR
      NMIS=NMIS-1
      IF(NMIS .GE. 0)GO TO 120
      GO TO 113
C
C        GENERATE COORDINATES FOR UNSPECIFIED INTERIOR NODES
C
  132 NELEM =N-1
      FACIT=1.3
C
C       DETERMINE WHICH ELEMENT SURROUND EACH NODE AND HOW MANY NODES
C   NEED TO BE GENERATED AND MAKE STARTING ESTIMATE FOR THEIR COORDINATES
C
      NMIS=0
      LOC=1
      DO 139 K=1,NPT
      IF(NQ(K) .LT. 0) GO TO 139
      NOC=0
      DO 137 I=1,NELEM
      DO 135 J=1,4
      JJ=J
  135 IF(K .EQ. NOD(I,J))GO TO 136
      GO TO 137
  136 N=10*I+JJ
      SL(LOC)=FLOAT(N)
      LOC=LOC+1
      NOC=NOC+1
  137 CONTINUE
      IF(NOC .GT. 0)GO TO 138
      NQ(K)=-1
      GO TO 139
  138 N=(LOC-NOC)*100-NOC
```

```
      NMIS=NMIS+6
      NQ(K)=N
      IF(K.LT.3) GO TO 139
      X(K)=0.5*(X(K-2)+X(K-1))
      Y(K)=0.5*(Y(K-2)+Y(K-1))
  139 CONTINUE
      IF(NMIS.EQ.0) GO TO 157
      WTL=0.0
C
C       ITERATE TO LOCATE UNSPECIFIED NODES
C       AS A WEIGHTED AVERAGE OF NEIGHBORS
C
      DO 155 NN=1,NMIS
      IOT=0
      DO 150 J=1,NPT
      N=NQ(J)
      IF(N .LT. 0) GO TO 150
      LOC=N/100
      NOC=MOD(N,100)
      WT=0.0
      XS=0.0
      YS=0.0
      DO 140 JJJ=1,NOC
      N=IFIX(SL(LOC)+0.1)
      LOC=LOC+1
      IE=N/10
      JJ=MOD(N,10)
      I=IS(JJ+3)
      I=NOD(IE,I)
      K=IS(JJ+1)
      K=NOD(IE,K)
      L=IS(JJ+2)
      L=NOD(IE,L)
      XS=XS          +X(I)    +X(K)    +X(L)*WTL
      YS=YS          +Y(I)    +Y(K)    +Y(L)*WTL
  140 WT=WT+2.0+WTL
      D1=X(J)
      D2=Y(J)
      X(J)=(1.0-FACIT)*D1+FACIT*XS/WT
      Y(J)=(1.0-FACIT)*D2+FACIT*YS/WT
      D1=ABS((X(J)-D1)/(ABS(D1)+1.0E-20))
      D2=ABS((Y(J)-D2)/(ABS(D2)+1.0E-20))
      IF(D1+D2 .GT. .0001)IOT=1
  150 CONTINUE
      IF(IOT .EQ. 0)  GO TO 157
      WTL=-1.
  155 CONTINUE
C
C       PRINT NODE AND ELEMENT DATA
C
  157 WRITE(6,908)
      WRITE(6,909)  (N,X(N),Y(N),N=1,NPT)
      WRITE(6,902)
      DO 168 N=1,NELEM
```

```
      MNORS=MNO(N)/100
      ISIGN=MOD(MNO(N),100)
      MNO(N)=MNORS
C
C   CHECK FOR NEGATIVE ELEMENT AREA, INITIALIZE STRESSES AND STRAINS,
C   PRINT INFORMATION, FORM ISOPARAMETRIC TRANSFORMATION AND STORE
C
      N1=NOD(N,1)
      N2=NOD(N,2)
      N3=NOD(N,3)
      N4=NOD(N,4)
      A1=X(N1)*(Y(N2)-Y(N4))+X(N2)*(Y(N4)-Y(N1))+X(N4)*(Y(N1)-Y(N2))
      A2=X(N2)*(Y(N3)-Y(N4))+X(N3)*(Y(N4)-Y(N2))+X(N4)*(Y(N2)-Y(N3))
      IF(A1+A2 .GT. 0.0) GO TO 166
      IX=0
      WRITE(6,920) N
      GO TO 167
  166 CALL GEOM(N,MNORS,ISIGN,SCOEF,SIGH,SIGV,U,XC,YC)
  167 WRITE(6,917) N,XC,YC,N1,N2,N3,N4,MNORS,SIGV,SIGH,U
  168 CONTINUE
      GO TO 10
C
C          NODE POINT SPECIFICATIONS
C
  175 WRITE (6,915)
      I=1
  180 READ(5,806) NSEC,KK,(IH(N),IIFLG(N),BIVD(N),N=1,2),TH,KKP,INCR,
     *            PJ,PK
      IF(NSEC .NE. 0)  GO TO 10
      IADD=0
      IF(MTYPE .EQ. 2)  IADD=4
      IF(TH .NE. 0.0)  IADD=8
      NMIS=0
      NM=1
      IF(KKP .EQ. 0)  GO TO 185
      IF(INCR .EQ. 0) INCR=1
      NM=(KKP-KK)/INCR
      NMIS=IABS(NM)
      INCR=INCR*NM/NMIS
      NM= 1 + NMIS
      DP=PK-PJ
      M=KK
      XL=0.0
C
C             GENERATE SPECIFICATIONS FOR INTERMEDIATE NODES
C
      DO 183 L=1,NMIS
      MP=M
      M=M+INCR
  183 XL=SQRT((X(M)-X(MP) )**2+(Y(M)-Y(MP) )**2)+XL
      DP=DP/XL
      RJ=1.0
      IF(MTYPE .EQ. 2)  RJ=X(KK)
  185 PXB=0.0
```

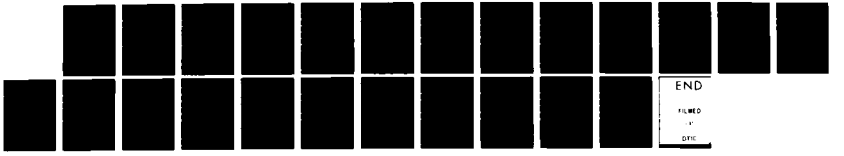
```
      PYB=0.0
      DO 190 L=1,NM
      PNB=0.0
      IF(L .EQ. NM) GO TO 186
      DX=X(KK + INCR) - X(KK)
      DY=Y(KK + INCR) - Y(KK)
      PK=PJ + DP*(SQRT(DX*DX + DY*DY))
      RK=1.0
      IF(MTYPE .EQ. 2)  RK=X(KK + INCR)
      PNB=(3.0*PJ*RJ + PJ*RK + PK*RJ + PK*RK)/12.0
      PNF=(PJ*RJ + PJ*RK + PK*RJ + 3.0*PK*RK)/12.0
C
C        CALCULATE EQUIVALENT NODAL FORCES DUE TO SPECIFIED PRESSURE
C
  186 BIV(I,1)=BIVD(1) + PXB - DY*PNB
      BIV(I,2)=BIVD(2) + PYB + DX*PNB
      PXB=-DY*PNF
      PYB= DX*PNF
      PJ=PK
      RJ=RK
      BIV(I,3)=TH
      NODE(I)=KK*1000000 + IH(1)*10000 + IH(2)*100
     *           + IIFLG(1)*10 + IIFLG(2)
      DO 187 J=1,2
      K=2*J-1+IADD
      AN(J)=ANI(K+1)
      IF(IIFLG(J) .EQ. 0)  GO TO 187
      AN(J)=ANI(K)
  187 CONTINUE
      WRITE(6,906) KK,(AN(J),BIV(I,J),HN(J),IH(J),J=1,2),AN(3),BIV(I,3)
      KK=KK+INCR
      BIV(I,3)=TH*0.0174533
      NBPTC=I
  190 I=I+1
      GO TO 180
C
C        THE SIZE OF THE PROBLEM IS CHECKED TO SEE IF IT IS TOO LARGE
C        AND DATA ERRORS ARE SOUGHT
C
  195 IF(NELEM .LT. 842)  GO TO 206
      WRITE (6,912)
      IX=0
  206 IF(NPT .LT. 901) GO TO 207
      WRITE (6,913)
      IX=0
  207 IF(NBPTC .LT. 121) GO TO 208
      WRITE (6,914)
      IX=0
  208 IF(NMAT .LT. 4) GO TO 210
      WRITE (6,911)
      IX=0
  210 RETURN
  700 STOP
      END
```

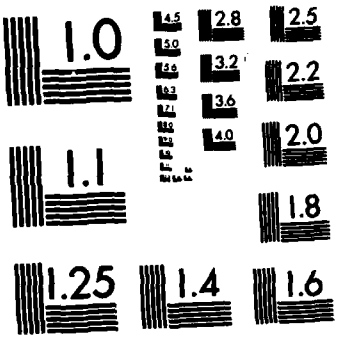
```
C*****************************************************************
      SUBROUTINE PROPTY(IN,MN,NMAT,NSEC)
C
C     THIS IS THE MASTER SUBROUTINE FOR SUPPLYING MATERIAL PROPERTIES
C
      COMMON/BLK0/ NPT,NELEM,NBPTC,MTYPE,IHISBF,ITMAX,RELAX,ERMAX,
     *             VITFAC,MODNEW,ITFAC,NONLIN,NWAY,IREPET,ITNO,NSTP
     *             ,ITIM,TIMF,TIMB,XKP,XXX
      COMMON/BLK1/ PROP(3,21),FXA(3),FYA(3),ITYPA(3)
      COMMON/BLK8/ C(4,4),SO(4),XPV(5),YPV(5),XJCOB(5),C1(8,8),ZY(8),
     *       FV(5,4),GV(5,4),XNV(5,4),SIGT(4),DSIG(4),EPT(4),DEP(4),
     *       STOR(6),PWPT,DPWP,CSC(4,4)
      COMMON/CNT/ ICNT1,ICNT2,ICNT3
      DIMENSION SIG3D(6),DSIG3D(6),EP3D(6),DEP3D(6),C3D(6,6),CB3D(6,6)
  801 FORMAT(8E10.3)
  807 FORMAT(I1,I4,I5,4E10.3)
  904 FORMAT(//,1X,'MATERIAL',I3,2X,'IS ISOTROPIC WITH FX(R)=',E10.3,4X,
     *  'FY(Z)=',E10.3,4X,'E =',E10.3,4X,'AND POISSONS RATIO =',F5.2,/)
  905 FORMAT(//,1X,'MATERIAL',I3,2X,'IS ANISOTROPIC WITH FX(R)=',1PE10.3
     *     , 4X, 'FY(Z)=',1PE10.3,4X,'AND',/,5X,'C11=',1PE10.3,
     *       5X,'C12=',0PE10.3,5X,'C13=',0PE10.3,5X,'C14=',0PE10.3,/,
     *      24X,'C22=',1PE10.3,5X,'C23=',0PE10.3,5X,'C24=',0PE10.3,/,
     *      43X,'C33=',1PE10.3,5X,'C34=',0PE10.3,/,
     *      62X,'C44=',1PE10.3)
C
      GO TO (100,300),IN
C
C     ********** READ MATERIALS PROPERTIES**********
C
  100 READ(5,807) NSEC,I,ITYP,FX,FY
      IF(NSEC .NE. 0)  RETURN
      IF(NMAT .LT. I)  NMAT=I
      FXA(I)=FX
      FYA(I)=FY
      ITYPA(I)=ITYP
      GO TO (150,160,200),ITYP
C
C     ISOTROPIC ELASTIC
C
  150 READ(5,801)E,GNU
      WRITE(6,904) I,FX,FY,E,GNU
      DU=E/((1.0+GNU) * (1.0-2.0*GNU))
      C11=DU*(1.0-GNU)
      C12=DU*GNU
      C13=C12
      C22=C11
      C23=C12
      C33=C11
      C44=E*0.5/(1.0+GNU)
      C14=0.0
      C24=0.0
      C34=0.0
      GO TO 165
C
```

```
C     ANISTROPIC ELASTIC
C
  160 READ(5,801)C11,C12,C13,C14,C22,C23,C24,C33,C34,C44
      WRITE(6,905) I,FX,FY,C11,C12,C13,C14,C22,C23,C24,C33,C34,C44
  165 PROP(I,1)=C11
      PROP(I,2)=C12
      PROP(I,3)=C13
      PROP(I,4)=C14
      PROP(I,5)=C22
      PROP(I,6)=C23
      PROP(I,7)=C24
      PROP(I,8)=C33
      PROP(I,9)=C34
      PROP(I,10)=C44
      GO TO 100
C
C     BOUNDING SURFACE PLASTICITY FOR COHESIVE SOIL
C
  200 CALL RPROP(PROP(I,1))
      GO TO 100
C
C     ***********CALCULATE INCREMENTAL PROPERTIES************
C
  300 ITYP=ITYPA(MN)
      GO TO (250,250,400),ITYP
C
C     LINEAR ELASTICITY
C
  250 IF(ITIM+ITNO .GT. 2)RETURN
      C(1,1)=PROP(MN,1)
      C(1,2)=PROP(MN,2)
      C(1,3)=PROP(MN,3)
      C(1,4)=PROP(MN,4)
      C(2,2)=PROP(MN,5)
      C(2,3)=PROP(MN,6)
      C(2,4)=PROP(MN,7)
      C(3,3)=PROP(MN,8)
      C(3,4)=PROP(MN,9)
      C(4,4)=PROP(MN,10)
      SO(1)=0.0
      SO(2)=0.0
      SO(3)=0.0
      SO(4)=0.0
      DO 280 J=1,4
      DO 280 K=1,J
  280 C(J,K)=C(K,J)
      RETURN
C
C     BOUNDING SURFACE MODEL FOR COHESIVE SOIL
C
  400 RT1=1.0
      RT2=1.0
C
C     CHANGE SIGN OF STRAIN ESTIMATE AT BEGINING OF NEW SOLUTION
```

AD-A124 866 NUMERICAL IMPLEMENTATION OF THE COHESIVE SOIL BOUNDING SURFACE PLASTICITY..(U) CALIFORNIA UNIV DAVIS DEPT OF CIVIL ENGINEERING L R HERRMANN ET AL. FEB 83 2/2

UNCLASSIFIED NCEL-CR-83.010 N62474-82-C-8276 F/G 8/13 NL





MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

```
C     ANISTROPIC ELASTIC
C
  160 READ(5,801)C11,C12,C13,C14,C22,C23,C24,C33,C34,C44
      WRITE(6,905) I,FX,FY,C11,C12,C13,C14,C22,C23,C24,C33,C34,C44
  165 PROP(I,1)=C11
      PROP(I,2)=C12
      PROP(I,3)=C13
      PROP(I,4)=C14
      PROP(I,5)=C22
      PROP(I,6)=C23
      PROP(I,7)=C24
      PROP(I,8)=C33
      PROP(I,9)=C34
      PROP(I,10)=C44
      GO TO 100
C
C     BOUNDING SURFACE PLASTICITY FOR COHESIVE SOIL
C
  200 CALL RPROP(PROP(I,1))
      GO TO 100
C
C     ***********CALCULATE INCREMENTAL PROPERTIES***********
C
  300 ITYP=ITYPA(MN)
      GO TO (250,250,400),ITYP
C
C     LINEAR ELASTICITY
C
  250 IF(ITIM+ITNO .GT. 2)RETURN
      C(1,1)=PROP(MN,1)
      C(1,2)=PROP(MN,2)
      C(1,3)=PROP(MN,3)
      C(1,4)=PROP(MN,4)
      C(2,2)=PROP(MN,5)
      C(2,3)=PROP(MN,6)
      C(2,4)=PROP(MN,7)
      C(3,3)=PROP(MN,8)
      C(3,4)=PROP(MN,9)
      C(4,4)=PROP(MN,10)
      SO(1)=0.0
      SO(2)=0.0
      SO(3)=0.0
      SO(4)=0.0
      DO 280 J=1,4
      DO 280 K=1,J
  280 C(J,K)=C(K,J)
      RETURN
C
C     BOUNDING SURFACE MODEL FOR COHESIVE SOIL
C
  400 RT1=1.0
      RT2=1.0
C
C     CHANGE SIGN OF STRAIN ESTIMATE AT BEGINING OF NEW SOLUTION
```

```
C     SEGMENT IN CASE OF UNSTABLE BEHAVIOR AT END OF PREVIOUS ONE
C
      IF(NSTP+ITNO .GT. 1)GO TO 405
      RT1=.01
      RT2=-.01
  405 DO 410 I=1,4
      SIG3D(I)=-SIGT(I)
      DSIG3D(I)=-DSIG(I)*RT1
      DSIG(I)=DSIG(I)*RT1
      EP3D(I)=-EPT(I)
      DEP3D(I)=-DEP(I)*RT2
  410 DEP(I)=DEP(I)*RT2
      DPWPT=DPWPT*RT1
      DO 415 I=5,6
      DSIG3D(I)=0.0
      SIG3D(I)=0.0
      EP3D(I)=0.0
  415 DEP3D(I)=0.0
      ITNOP=ITNO
      LITNO=0
  420 LITNO=LITNO+1
      ITNOP=ITNOP+1
C************NOTE**********NOTE****
      ICNT3=ICNT3+1
      CALL CLAY(3,ITIM,ITNOP,PROP(MN,1),STOR,SIG3D,EP3D,DSIG3D,DEP3D,
     1          C3D,CB3D,DPWPT,DPWPT,GAM,1,0)
      R1=.5
      IF(NONLIN .EQ.2)R1=1.0
      R2=1.0-R1
      IF(NONLIN .EQ.1  .AND.  MODNEW .EQ. 1)GO TO 460
      DO 450 I=1,4
      SO(I)=0.0
      DO 450 J=1,4
      CSC(I,J)=0.5*(C3D(I,J)+CB3D(I,J))
  450 C(I,J)=C3D(I,J)*R1+CB3D(I,J)*R2
  460 IF(NWAY .LT. 1)GO TO 625
      DO 560 I=1,4
      DU=DSIG(I)
      DO 550 J=1,4
  550 DU=DU-0.5*(C3D(I,J)+CB3D(I,J))*DEP(J)
  560 SO(I)=DU
      ER=0.0
      XNRM=0.0
      DO 600 I=1,4
      ER=ER+ABS(SO(I))
      XNRM=XNRM+ABS(DSIG(I))
      DSIG(I)=DSIG(I)-SO(I)
  600 DSIG3D(I)=-DSIG(I)
      IF(XNRM .EQ. 0.0 )GO TO 625
      IF(LITNO .EQ. ITMAX)GO TO 620
      IF(ER/XNRM .GT. 10.*ERMAX)GO TO 420
  620 DU=ER/XNRM
  625 IF(MODNEW .EQ. 0  .OR.  NONLIN .EQ. 2)GO TO 629
      DO 628 I=1,4
```

```
      SO(I)=0.0
      DO 628 J=1,4
  628 SO(I)=SO(I)+(0.5*C3D(I,J)+0.5*CB3D(I,J)-C(I,J))*DEP(J)
      RETURN
C**********NOTE**********NOTE
  629 R1=0.5
      DO 630 I=1,4
      SO(I)=0.0
      DO 630 J=1,4
  630 SO(I)=SO(I)+(R1-0.5)*(C3D(I,J)+CB3D(I,J))*DEP(J)
      RETURN
      END
C ***************************************************************************
      SUBROUTINE STRESS(ITSTOP)
C
C     THIS SUBROUTINE CALCULATES AND PRINTS ELEMENT STRESSES AND STRAINS
C
      COMMON/BLK0/ NPT,NELEM,NBPTC,MTYPE,IHISBF,ITMAX,RELAX,ERMAX,
     *             VITFAC,MODNEW,ITFAC,NONLIN,NWAY,IREPET,ITNO,NSTP
     *             ,ITIM,TIMF,TIMB,XKP,XXX
      COMMON/BLK2/ X(900),Y(900),NQ(901),DISPLT(1800)
      COMMON/BLK3/ MNO(841),NOD(841,4)
      COMMON/BLK5/ SL(1800),SLP(1800),SLPP(1800)
      COMMON/BLK8/ C(4,4),SO(4),XPV(5),YPV(5),XJCOB(5),C1(8,8),ZY(8),
     *      FV(5,4),GV(5,4),XNV(5,4),SIGT(4),DSIG(4),EPT(4),DEP(4),
     *      STOR(6),PWPT,DPWP,CSC(4,4)
      DIMENSION U(2),UX(2),UY(2)
C
C     FORMAT STATEMENTS
C
  920 FORMAT(I8,3X,1P9E12.3)
  922 FORMAT(//,5X,7HELEMENT,45X,
     1          28HELEMENT STRAINS AND STRESSES,/,6X,3HNO.,5X,
     2           9HEPSILON-X,3X,9HEPSILON-Y,3X,9HEPSILON-Z
     3           ,3X,8HGAMMA-XY,4X,7HSIGMA-X,5X,7HSIGMA-Y,5X,7HSIGMA-Z,
     4           5X,6HTAU-XY,10X,'U')
  923 FORMAT(//,5X,'ELEMENT',45X,
     1           'ELEMENT STRAINS AND STRESSES',/,6X,'NO.',5X,
     2           'EPSILON-R',3X,'EPSILON-Z',3X,'EPS-THETA'
     3      ,3X,'GAMMA-RZ',4X,'SIGMA-R',5X,'SIGMA-Z',4X,'SIG-THETA',
     4       4X,'TAU-RZ',10X,'U')
  924 FORMAT(1HO,4X,4HNODE, 7X, 13HDISPLACEMENTS, / 6X, 3HNO. , 8X,
     1          1HU, 10X, 1HV)
C
C     FOR EACH ELEMENT FIND STRAINS AND STRESSES
C
      IF(ITSTOP .EQ. 0)  GO TO 110
      IF(MTYPE .LT. 2)  WRITE(6,922)
      IF(MTYPE .EQ. 2)  WRITE(6,923)
  110 DO 760 IX=1,NELEM
      MN=MNO(IX)
      NOD(IX,1)=IABS(NOD(IX,1))
C
C      RECALL ELEMENT INFORMATION FROM DISK
```

```
C
      READ(2=IX)((SO(J),(C(I,J),CSC(I,J),I=1,4),J=1,4),(XPV(K),YPV(K),
     *      XJCOB(K),(FV(K,L),GV(K,L),XNV(K,L),L=1,4),K=1,5),(SIGT(M),
     *          DSIG(M),EPT(M),DEP(M),M=1,4),PWPT,DPWP,(STOR(N),N=1,6))
C
C             CALCULATE THE STRESS AND STRAIN AT THE ELEMENT CENTER
C
      DO 157 J=1,2
      DU1=0.0
      DU2=0.0
      DU3=0.0
      DO 155 I=1,4
      NN=NOD(IX,I)
      II=NQ(NN)+J-1
C     **********NOTE*******NOTE******NOTE
      UN=SL(II)
      DU1=DU1 + XNV(5,I)*UN
      DU2=DU2 + FV(5,I)*UN
  155 DU3=DU3 + GV(5,I)*UN
      U(J)=DU1
      UX(J)=DU2
  157 UY(J)=DU3
      RO=0.0
      IF(MTYPE .EQ. 2)  RO=1.0/XPV(5)
      DEP(1)=UX(1)
      DEP(2)=UY(2)
      DEP(4)=UX(2) + UY(1)
      DEP(3)=RO*U(1)-XKP*(CSC(1,3)*DEP(1)+CSC(2,3)*DEP(2)
     1      + CSC(3,4)*DEP(4)+ SO(3))/CSC(3,3)
      DO 400 I=1,4
      DU=SO(I)
C***************NOTE************NOTE*****
      IF(NONLIN .EQ. 2)DU=0.0
      DO 380 J=1,4
  380 DU=DU + CSC(I,J)*DEP(J)
  400 DSIG(I)=DU
      DSIG(3)=DSIG(3)*(1.0 - XKP)
      DPWP=0.0
      IF(ITSTOP .EQ. 0)  GO TO 750
C
C      IF CONVERGENCE HAS OCCURRED, SUM ELEMENT STRESSES AND STRAINS
C      AND PRINT RESULTS
C
      DO 740 J=1,4
      EPT(J)=EPT(J) + DEP(J)
  740 SIGT(J)=SIGT(J) + DSIG(J)
      PWPT=PWPT + DPWP
      WRITE(6,920) IX,(EPT(J),J=1,4),(SIGT(J),J=1,4),PWPT
  750 CONTINUE
      WRITE(2=IX)((SO(J),(C(I,J),CSC(I,J),I=1,4),J=1,4),(XPV(K),YPV(K),
     *      XJCOB(K),(FV(K,L),GV(K,L),XNV(K,L),L=1,4),K=1,5),(SIGT(M),
     *          DSIG(M),EPT(M),DEP(M),M=1,4),PWPT,DPWP,(STOR(N),N=1,6))
  760 CONTINUE
      IF(ITSTOP .EQ. 0)  RETURN
```

```
C
C     DISPLACEMENTS SUMMED AND PRINTED
C
      WRITE(6,924)
      DO 785 I=1,NPT
      JJ=NQ(I)
      IF(JJ .EQ. NQ( I+1))GO TO 785
      DISPLT(JJ)=DISPLT(JJ) + SL(JJ)
      DISPLT(JJ+1)=DISPLT(JJ+1) + SL(JJ+1)
      WRITE(6,920) I,DISPLT(JJ),DISPLT(JJ+1)
  785 CONTINUE
      RETURN
      END
C ***************************************************************************
      SUBROUTINE STIFNS(IX,LSTND,LT,NCOL,S)
C
C     THIS SUBROUTINE FORMS THE ELEMENT MATRIX FOR A QUADRILATERAL
C     ELEMENT
C
      COMMON/BLK0/ NPT,NELEM,NBPTC,MTYPE,IHISBF,ITMAX,RELAX,ERMAX,
     *             VITFAC,MODNEW,ITFAC,NONLIN,NWAY,IREPET,ITNO,NSTP
     *             ,ITIM,TIMF,TIMB,XKP,XXX
      COMMON/BLK1/ PROP(3,21),FXA(3),FYA(3),ITYPA(3)
      COMMON/BLK2/ X(900),Y(900),NQ(901),DISPLT(1800)
      COMMON/BLK3/ MNO(841),NOD(841,4)
      COMMON/BLK4/ NODB(120),BIV(120,3)
      COMMON/BLK5/ SL(1800),SLP(1800),SLPP(1800)
      COMMON/BLK8/ C(4,4),SO(4),XPV(5),YPV(5),XJCOB(5),C1(8,8),ZY(8),
     *      FV(5,4),GV(5,4),XNV(5,4),SIGT(4),DSIG(4),EPT(4),DEP(4),
     *      STOR(6),PWPT,DPWP,CSC(4,4)
      DIMENSION IIFLG(2),BIVD(2),S(LT,NCOL),C2(8,8)
C
C      RECALL ELEMENT INFORMATION FROM DISK
C
      READ(2=IX)((SO(J),(C(I,J),CSC(I,J),I=1,4),J=1,4),(XPV(K),YPV(K),
     *      XJCOB(K),(FV(K,L),GV(K,L),XNV(K,L),L=1,4),K=1,5),(SIGT(M),
     *          DSIG(M),EPT(M),DEP(M),M=1,4),PWPT,DPWP,(STOR(N),N=1,6))
C
C     CALCULATE INCREMENTAL PROPERTIES
C
      MN=MNO(IX)
      IN=2
      CALL PROPTY(IN,MN,NZ,NZ)
      C11=C(1,1)
      C12=C(1,2)
      C13=C(1,3)
      C14=C(1,4)
      C22=C(2,2)
      C23=C(2,3)
      C24=C(2,4)
      C33=C(3,3)
      C34=C(3,4)
      C44=C(4,4)
      C11X=CSC(1,1)
```

```
      C12X=CSC(1,2)
      C13X=CSC(1,3)
      C14X=CSC(1,4)
      C22X=CSC(2,2)
      C23X=CSC(2,3)
      C24X=CSC(2,4)
      C33X=CSC(3,3)
      C34X=CSC(3,4)
      C44X=CSC(4,4)
      S1=SO(1)
      S2=SO(2)
      S3=SO(3)
      S4=SO(4)
C
C              INITIALIZE ELEMENT MATRICES
C
      DO 60 J=1,8
      DO 50 K=1,8
      C2(J,K)=0.0
   50 C1(J,K)=0.0
   60 ZY(J)=0.0
C
C     SET PARAMETERS FOR TYPE OF 2-D ANALYSIS TO BE PERFORMED
C
      R=1.0
      RO=0.0
C
C          PLANE STRESS TERMS
C
      D11=XKP*C13*C13/C33
      D12=XKP*C13*C23/C33
      D14=XKP*C13*C34/C33
      D22=XKP*C23*C23/C33
      D24=XKP*C23*C34/C33
      D44=XKP*C34*C34/C33
      CALL INTP(IHISBF,TIMB,TIMF,ITIM,DF)
      MN=MNO(IX)
      FX=FXA(MN)*DF
      FY=FYA(MN)*DF
C
C           NUMERICAL INTEGRATION LOOP
C
      DO 240 N=1,4
      IF(MTYPE .LT. 2)  GO TO 152
      R=XPV(N)
      RO=1.0/R
C
C          CALCULATE ELEMENT MATRICES
C
  152 DU=XJCOB(N)*R
      DO 220 I=1,4
      I1=2*I-1
      I2=I1+1
      FI=FV(N,I)
```

```
      GI=GV(N,I)
      XNI=XNV(N,I)
C
C        ELEMENT LOAD MATRIX
C
      ZY(I1)=ZY(I1) + DU*(XNI*(FX-RO*S3) - FI*(S1-XKP*C13/C33*S3)
     1              - GI*(S4 - XKP*C34/C33*S3))
      ZY(I2)=ZY(I2) + DU*(XNI*FY - FI*(S4-XKP*C34/C33*S3)
     1              - GI*(S2 - XKP*C23/C33*S3))
      IF(MODNEW .EQ. 1)  GO TO 220
      DO 210 J=I,4
      J1=2*J-1
      J2=J1+1
      FJ=FV(N,J)
      GJ=GV(N,J)
      XNJ=XNV(N,J)
C
C     ELEMENT STIFFNESS MATRIX
C
      C2(I1,J1)=C2(I1,J1) + DU*((C11X)*FI*FJ + C33X*RO*RO*XNI*XNJ
     1          + (C44X)*GI*GJ + C13X*RO*(FI*XNJ + FJ*XNI)
     2          + (C14X)*(GI*FJ + GJ*FI) + C34X*RO*(XNI*GJ+XNJ*GI))
      C2(I1,J2)=C2(I1,J2) + DU*((C44X)*GI*FJ + (C12X)*FI*GJ
     1          + (C14X)*FI*FJ + C23X*RO*XNI*GJ + (C24X)*GI*GJ
     2          + C34X*RO*XNI*FJ)
      C2(I2,J1)=C2(I2,J1) + DU*((C44X)*FI*GJ + (C12X)*FJ*GI
     1          + (C14X)*FI*FJ + C23X*RO*GI*XNJ + (C24X)*GI*GJ
     2          + C34X*RO*XNJ*FI)
      C2(I2,J2)=C2(I2,J2) + DU*((C22X)*GI*GJ + (C44X)*FI*FJ
     1          + (C24X)*(FI*GJ + FJ*GI))
      C1(I1,J1)=C1(I1,J1) + DU*((C11-D11)*FI*FJ + C33*RO*RO*XNI*XNJ
     1          + (C44-D44)*GI*GJ + C13*RO*(FI*XNJ + FJ*XNI)
     2          + (C14-D14)*(GI*FJ + GJ*FI) + C34*RO*(XNI*GJ+XNJ*GI))
      C1(I1,J2)=C1(I1,J2) + DU*((C44-D44)*GI*FJ + (C12-D12)*FI*GJ
     1          + (C14-D14)*FI*FJ + C23*RO*XNI*GJ + (C24-D24)*GI*GJ
     2          + C34*RO*XNI*FJ)
      C1(I2,J1)=C1(I2,J1) + DU*((C44-D44)*FI*GJ + (C12-D12)*FJ*GI
     1          + (C14-D14)*FI*FJ + C23*RO*GI*XNJ + (C24-D24)*GI*GJ
     2          + C34*RO*XNJ*FI)
  210 C1(I2,J2)=C1(I2,J2) + DU*((C22-D22)*GI*GJ + (C44-D44)*FI*FJ
     1          + (C24-D24)*(FI*GJ + FJ*GI))
  220 CONTINUE
  240 CONTINUE
      DO 245 J=1,8
      DO 245 I=1,8
      C2(I,J)=C2(J,I)
  245 C1(I,J)=C1(J,I)
      IF(NONLIN .EQ. 1)GO TO 270
      DO 260 I=1,8
      DU=0.0
      J=0
      DO 250 JJ=1,4
      DO 250 N=1,2
      L=NOD(IX,JJ)
```

```
      L=NQ(L)+N-1
      J=J+1
  250 DU=DU+C2(I,J)*SLP(L)
  260 ZY(I)=-DU
  270 CONTINUE
C
C              THE NODE POINT SPECIFICATIONS ARE CONSIDERED
C
      NU=2
      DO 334 J=1,4
      NRQ=NOD(IX,J)
      KL=NBPTC + 1
      DO 331 K=1,KL
      NR=NU*(J-1)
      IF(K .LT. KL)  GO TO 300
C
C            APPLY BOUNDARY CONDITION WHEN R=0
C
      IF(MTYPE .LT. 2 .OR. X(NRQ) .NE. 0.0)  GO TO 331
      IIFLG(1)=1
      IIFLG(2)=0
      BIVD(1)=0
      BIVD(2)=0
      GO TO 320
  300 KK=IABS(NODB(K))
      K1=KK/1000000
      IF (NRQ .NE. K1) GO TO 331
      XX=1.0
      IF (NODB(K) .LT. 0) XX=0.0
      NODB(K)=-KK
      IIFLG(1)=MOD(KK,100)/10
      IIFLG(2)=MOD(KK,10)
      IH1=MOD(KK,1000000)/10000
      CALL INTP(IH1,TIMB,TIMF,ITIM,DF)
      BIVD(1)=BIV(K,1)*DF
      IH2=MOD(KK,10000)/100
      CALL INTP(IH2,TIMB,TIMF,ITIM,DF)
      BIVD(2)=BIV(K,2)*DF
      ANG=BIV(K,3)
      IF(ANG .EQ. 0.0) GO TO 320
C
C              TRANSFORMATION TO LOCAL COORDINATE AXES
C
      CA=COS(ANG)
      SA=SIN(ANG)
      D1=ZY(NR+1)
      D2=ZY(NR+2)
      ZY(NR+1)=D1*CA+D2*SA
      ZY(NR+2)=-D1*SA+D2*CA
      DO 315 JJ=1,8
      D1=C1(NR+1,JJ)
      D2=C1(NR+2,JJ)
      C1(NR+1,JJ)=D1*CA+D2*SA
  315 C1(NR+2,JJ)=-D1*SA+D2*CA
```

```
      DO 318 II=1,8
      D1=C1(II,NR+1)
      D2=C1(II,NR+2)
      C1(II,NR+1)=D1*CA+D2*SA
  318 C1(II,NR+2)=-D1*SA+D2*CA
  320 DO 330 N=1,2
      DU1=BIVD(N)
      NR=NR+1
C
C        LOADS ADDED IN
C
      ZY(NR)=ZY(NR)+XX*DU1
      IF(IIFLG(N)   .EQ. 0) GO TO 330
C
C        DISPLACEMENTS SPECIFIED
C
C*****************NOTE*****************NOTE
      IF(NONLIN .EQ. 1)GO TO 325
      NRM=NQ(NRQ)-1+N
      DU1=DU1-SLP(NRM)
  325 ZY(NR)=DU1 * XX * XXX
      C1(NR,NR)=XXX * XX
  330 CONTINUE
  331 CONTINUE
  334 CONTINUE
C
C       THE ELEMENT MATRIX IS NOW ADDED INTO THE SYSTEM MATRIX
C
      NRCC=0
      DO 355 K=1,4
      KK=NOD(IX,K)
      NR=NQ(KK)-1
      DO 350 M=1,NU
      NRCC=NRCC+1
      NR=NR + 1
      IF(MODNEW .EQ. 1)  GO TO 350
      NRM=NR - LSTND

      NCCC=0
      DO 345 L=1,4
      JJ=NOD(IX,L)
      NCN=NQ(JJ) - NR
      DO 345 N=1,NU
      NCCC=NCCC+1
      NCN=NCN+1
      IF (NCN.LT. 1) GO TO 344
      S(NRM,NCN)=S(NRM,NCN) + C1(NRCC,NCCC)
  344 CONTINUE
  345 CONTINUE
  350 SL(NR)=SL(NR) + ZY(NRCC)
  355 CONTINUE
      WRITE(2=IX)((SO(J),(C(I,J),CSC(I,J),I=1,4),J=1,4),(XPV(K),YPV(K),
     *      XJCOB(K),(FV(K,L),GV(K,L),XNV(K,L),L=1,4),K=1,5),(SIGT(M),
     *          DSIG(M),EPT(M),DEP(M),M=1,4),PWPT,DPWP,(STOR(N),N=1,6))
```

```
      RETURN
      END
C ***********************************************************************
      SUBROUTINE SOLVE (L1,L2,LT,NROW,NCOL,IDISK,Q7,S)
C
C       THIS SUBROUTINE FORMS AND SOLVES THE SIMULTANEOUS EQUATIONS.
C       VARIABLE DIMENSIONING IS USED TO MAXIMIZE THE LENGTH OF THE MAIN
C       BLOCK. DISK STORAGE IS USED WHEN MORE THAN ONE BLOCK IS REQUIRED.
C       WHEN MODNEW=1, ONLY REDUCTION OF THE RIGHTHAND SIDE AND BACK
C       SUBSTITUTION TAKES PLACE.
C
      COMMON/BLK0/ NPT,NELEM,NBPTC,MTYPE,IHISBF,ITMAX,RELAX,ERMAX,
     *             VITFAC,MODNEW,ITFAC,NONLIN,NWAY,IREPET,ITNO,NSTP
     *             ,ITIM,TIMF,TIMB,XKP,XXX
      COMMON/BLK2/ X(900),Y(900),NQ(901),DISPLT(1800)
      COMMON/BLK3/ MNO(841),NOD(841,4)
      COMMON/BLK5/ SL(1800),SLP(1800),SLPP(1800)
      DIMENSION S(LT,NCOL),Q7(1)
C
C         THE STIFFNESS MATRIX IS GENERATED IN BLOCKS AND STORED ON DISK
C
      ID=0
  288 ID=ID + 1
      LSTND=(ID-1)*L1
      NWND=LSTND + L1
C
C         EACH ELEMENT IS EXAMINED TO DETERMINE IF IT CONTRIBUTES
C         TO THE BLOCK
C
      DO  355 I=1,NELEM
      IX=I
      IF(NOD(I,1) .LT. 0) GO TO 355
      DO 300 L=1,4
      KK=NOD(I,L)
      K1=NQ(KK)
      IF ( K1 .LE. NWND)  GO TO 305
  300 CONTINUE
      GO TO 355
C
C         CALCULATE THAT PORTION OF THE STIFFNESS MATRIX GIVEN BY A
C         CONSIDERATION OF ELEMENT I
C
  305 CALL STIFNS (IX,LSTND,LT,NCOL,Q7)
      NOD(I,1)=-NOD(I,1)
  355 CONTINUE
C
C         THE BLOCK OF EQUATIONS IS REDUCED AND PUT ON DISK IF REQUIRED
C
      IF(MODNEW .EQ. 0 )  GO TO 400
      IF(IDISK .NE. 0) READ(1=ID) ((S(N,M),M=1,NCOL),N=1,L1)
      GO TO 505
C
C         PRODUCE THE LEFTHAND SIDE
C
```

```
  400 DO 500 N=1,L1
      DIAG=S(N,1)
      IF(DIAG .EQ. 0.0)  GO TO 500
      I=N
      DO 475 L=2,NCOL
      C=S(N,L)/DIAG
      I=I+1
      IF(C .EQ. 0.0)  GO TO 475
      J=0
      DO 450 K=L,NCOL
      J=J+1
  450 S(I,J)=S(I,J) - C*S(N,K)
  475 CONTINUE
  500 CONTINUE
C******FOR LINEAR SYSTEMS INCLUDE FOLLOWING STATEMENT
C     IF(ID .GE. IDISK)GO TO 505
      IF(IDISK .GT. 0)  WRITE(1=ID) ((S(N,M),M=1,NCOL),N=1,L1)
C
C         REDUCE THE RIGHTHAND SIDE
C
  505 DO 520 N=1,L1
      DIAG=S(N,1)
      IF(DIAG .EQ. 0.0)  GO TO 520
      NR=N + LSTND
      D=SL(NR)/DIAG
      I=NR
      DO 510 L=2,NCOL
      I=I+1
C********IF CANNOT OVER FLOW SL IN COMMON, INCLUDE NEXT STATEMENT
      IF(I .GT. NROW)GO TO 520
  510 SL(I)=SL(I) - S(N,L)*D
  520 CONTINUE
      IF(ID .GE. IDISK .OR. MODNEW .EQ. 1)  GO TO 600
C
C        THE NUMBER TWO BLOCK OF EQUATIONS IS SHIFTED INTO THE
C        NUMBER ONE POSITION
C
      N=L1
      DO 530 I=1,L2
      N=N+1
      DO 530 J=1,NCOL
      S(I,J)=S(N,J)
  530 S(N,J)=0.0
      IF(L2 .GE. L1)  GO TO 600
      LL=L2 + 1
      DO 540 I=LL,L1
      DO 540 J=1,NCOL
  540 S(I,J)=0.0
  600 IF(ID .LT. IDISK)  GO TO 288
C
C        BACK SUBSTITUTION
C
      ID=IDISK + 1
      NR=IDISK*L1+1
```

```
      IF(IDISK .EQ. 0)NR=NROW+1
  610 ID=ID - 1
      IF(ID .LT. IDISK)  READ(1=ID) ((S(N,M),M=1,NCOL),N=1,L1)
      N=L1 + 1
      DO 750 M=1,L1
      NR=NR - 1
      N=N -1
      IF(NR .GT. NROW)GO TO 750
      DU1=SL(NR)
      IF(S(N,1) .EQ. 0.0)  GO TO 750
      L=NR
      DO 725 K=2,NCOL
      L=L + 1
C********IF CANNOT OVER FLOW SL IN COMMON, INCLUDE NEXT STATEMENT
      IF(L .GT. NROW)GO TO 727
  725 DU1=DU1-S(N,K)*SL(L)
  727 SL(NR)=DU1/S(N,1)
  750 CONTINUE
      IF(ID .GT. 1)  GO TO 610
      RETURN
      END
C  ***************************************************************
      SUBROUTINE INTP (I,T2,T1,ITIM,DF)
C
C         SUBROUTINE TO INTERPOLATE HISTORY FUNCTION
C
      COMMON/BLK7/ FUN(10,3),FUNT(10,3),NPTS(3,3)
      DIMENSION TT(2),F(2)
      DF=1.0
      IF(I .LT. 0)  RETURN
      IF(I .GT. 0)  GO TO 20
      IF(ITIM .GT. 1)  DF=0.0
      RETURN
   20 DF=FUNT(1,I)
      IF(ITIM .EQ. NPTS(I,1))  RETURN
      NPTS(I,1)=ITIM
      TT(2)=T2
      TT(1)=T1
      NP=NPTS(I,2)
      N=NPTS(I,3)
      FUNT(1,I)=0.0
      IF(TT(2) .LT. FUNT(NP,I))  NP=1
      DO 300 LL=1,2
      L=3-LL
      T=TT(L)
      DO 100 J=NP,N
      K=J
      IF(T .LE. FUNT(K+1,I))  GO TO 200
  100 CONTINUE
  200 F(L)=FUN(K,I) + (FUN(K+1,I)-FUN(K,I))*(T-FUNT(K,I))/
     *     (FUNT(K+1,I) - FUNT(K,I))
  300 CONTINUE
      NPTS(I,2)=K
      DF=F(1) - F(2)
```

```
      FUNT(1,I)=DF
      RETURN
      END
      SUBROUTINE GEOM(IX,MN,ISIGN,SCOEF,SIGH,SIGV,U,XC,YC)
C*****************************************************************
C
C   THIS SUBROUTINE INITIALIZES THE STRESS AND PROPERTY ARRAYS AND
C   CALCULATES THE ISOPARAMETRIC TRANSFORMATION AND STORES ON DISK
C
      COMMON/BLK2/ X(900),Y(900),NQ(901),DISPLT(1800)
      COMMON/BLK3/ MNO(841),NOD(841,4)
      COMMON/BLK6/ ROA(4),SCA(4),ETA(4)
      COMMON/BLK8/ C(4,4),SO(4),XPV(5),YPV(5),XJCOB(5),C1(8,8),ZY(8),
     *      FV(5,4),GV(5,4),XNV(5,4),SIGT(4),DSIG(4),EPT(4),DEP(4),
     *      STOR(6),PWPT,DPWP,CSC(4,4)
      DIMENSION SCOEF(8,6),RQ(4),ZQ(4)
C
C        THE AND NODE POINT COORDINATES ARE FOUND
C
      DO 100 J=1,4
      K=NOD(IX,J)
      RQ(J)=X(K)
  100 ZQ(J)=Y(K)
C
C              ISOPARAMETRIC TRANSFORMATION FACTORS
C
      A1=ZQ(1)+ZQ(2)-ZQ(3)-ZQ(4)
      A2=ZQ(1)-ZQ(2)-ZQ(3)+ZQ(4)
      B= ZQ(1)-ZQ(2)+ZQ(3)-ZQ(4)
      C1P=RQ(1)-RQ(2)-RQ(3)+RQ(4)
      C2=RQ(1)+RQ(2)-RQ(3)-RQ(4)
      D= RQ(1)-RQ(2)+RQ(3)-RQ(4)
C
C            QUADRATURE POINT LOOP
C
      DO 155 N=1,5
      IF(N .LT. 5)  GO TO 111
      SC=0.0
      ET=0.0
      GO TO 114
  111 DU=1.0/SQRT(3.0)
      SC=DU*SCA(N)
      ET=DU*ETA(N)
  114 D1=C1P+D*ET
      D2=A1+B*SC
      D3=C2+D*SC
      D4=A2+B*ET
      DU=1.0/(D1*D2-D3*D4)
C
C              CALCULATION OF SHAPE FUNCTION DERIVATIVES
C
      XC=0.0
      YC=0.0
      DO 150 I=1,4
```

```
      D5=ROA(I)
      D6=SCA(I)
      D7=ETA(I)
      FV(N,I)=DU*(D2*(D6 + D5*ET) - D4*(D7 + D5*SC))
      GV(N,I)=DU*(D1*(D7 + D5*SC) - D3*(D6 + D5*ET))
      D8=0.25*(1.0 + D6*SC)*(1.0 + D7*ET)
      XNV(N,I)=D8
      XC=XC + RQ(I)*D8
  150 YC=YC + ZQ(I)*D8
      XPV(N)=XC
      YPV(N)=YC
      XJCOB(N)=0.0625/DU
  155 CONTINUE
C
C       INITIALIZE STRESSES AND STRAINS
C
      U=0.0
      SIGH=0.0
      SIGV=0.0
      DPWP=0.0
      DO 160 I=1,4
      DSIG(I)=0.0
      DEP(I)=0.0
  160 EPT(I)=0.0
      IF(ISIGN .EQ. 0)  GO TO 162
      SIGV=SCOEF(ISIGN,1) + SCOEF(ISIGN,2)*YC
      SIGH=SCOEF(ISIGN,3) + SCOEF(ISIGN,4)*YC
      U=SCOEF(ISIGN,5) + SCOEF(ISIGN,6)*YC
  162 SIGT(1)=SIGH
      SIGT(2)=SIGV
      SIGT(3)=SIGH
      PWPT=U
      SIGT(4)=0.0
C
C        STORE INFORMATION ON DISK
C
      WRITE(2=IX)((SO(J),(C(I,J),CSC(I,J),I=1,4),J=1,4),(XPV(K),YPV(K),
     *      XJCOB(K),(FV(K,L),GV(K,L),XNV(K,L),L=1,4),K=1,5),(SIGT(M),
     *          DSIG(M),EPT(M),DEP(M),M=1,4),PWPT,DPWP,(STOR(N),N=1,6))
      RETURN
      END
C*****************************************************************************
      SUBROUTINE RPROP(PROP)
C
C     THIS SUBROUTINE READS IN AND SCALES THE PROPERTIES REQUIRED
C     BY THE BOUNDING SURFACE PLASTICITY MODEL FOR COHESIVE SOILS.
C
      DIMENSION PROP(21)
      READ(5,801) (PROP(I),I=1,3),(PROP(I),I=9,11),PROP(7),
     1            PROP(21),PROP(17),PROP(16),PROP(20),PROP(5),
     2            PROP(6),PROP(8),PROP(4),PROP(18),PROP(15),
     3            PROP(12),PROP(13),PROP(19),PROP(14)
  801 FORMAT(8E10.3)
      WRITE(6,901) (PROP(I),I=1,3),PROP(4)
```

```
  901 FORMAT(5X,'CLAY PROPERTIES'//15X,'LAMBDA =',E10.3/15X,
     1       'KAPPA  =',E10.3/15X,'MC     =',E10.3/15X,
     2       'ME/MC  =',E10.3)
      WRITE(6,902) PROP(9),PROP(12),PROP(10),PROP(13),
     1             PROP(11),PROP(7),PROP(21),PROP(17),
     1             PROP(19),PROP(16),PROP(18),
     1             PROP(14),PROP(20)
  902 FORMAT(15X,'RC     =',E10.3,15X,'RE/RC   =',E10.3/
     1       15X,'AC     =',E10.3,15X,'AE/AC   =',E10.3/
     2       15X,'T      =',E10.3,15X,'PL      =',E10.3/
     3       15X,'PO     =',E10.3/15X,'HARDENING PARAMETERS:'/
     4       19X,'SMC     =',E10.3,10X,'SME/SMC =',E10.3/
     5       19X,'HC      =',E10.3,10X,'HE/HC   =',E10.3/
     6       15X,'PROJECTION POINT 1/C =',E10.3/
     7       15X,'INITIAL VOID RATIO   =',E10.3)
      IF(PROP(5).LT.0.5) WRITE(6,903) PROP(5)
      IF(PROP(5).GE.0.5) WRITE(6,904) PROP(5)
  903 FORMAT(15X,'POISSON*S RATIO      =',E10.3)
  904 FORMAT(15X,'SHEAR MODULUS        =',E10.3)
      WRITE(6,905) PROP(8)
  905 FORMAT(15X,'ATMOSPHERIC PRESSURE =',E10.3)
      WRITE(6,906) PROP(15)
  906 FORMAT(15X,'SIZE OF ELASTIC ZONE =',E10.3//)
      IF(PROP(6).EQ.0.0) WRITE(6,907)
  907 FORMAT(5X,'***** DRAINED CONDITIONS *****'//)
      IF(PROP(6).NE.0.0) WRITE(6,908) PROP(6)
  908 FORMAT(5X,'***** UNDRAINED CONDITIONS --- THE COMBINED ',
     1       'SKELETON AND WATER BULK MODULUS =',E10.3//)
      PROP(3)=PROP(3)/(3.0*SQRT(3.0))
      PROP(7)=PROP(7)*3.0
      PROP(21)=PROP(21)*3.0
      RETURN
      END
C*****************************************************************
      SUBROUTINE CLAY(IDIM,INC,ITNO,PROP,STOR,SIGBM,EPM,
     1                DSIGM,DEPM,C,CB,UB,DLTAU,GAM,KIND,LARGE)
C
C     SUBROUTINE TO EVALUATE YANNIS DAFALIAS' BOUNDING
C     SURFACE PLASTICITY MODEL FOR CLAY SOILS.  PREPARED BY
C     L.R. HERRMANN AT THE UNIVERSITY OF CALIFORNIA, DAVIS CAMPUS.
C
      DIMENSION PROP(21),STOR(6),SIGB(6),DSIG(6),DEP(6),C(6,6),
     1          SB(3,3),SF(3,3),II(6),DLTA(3,3),DEPM(6),
     2          SIGBM(6),DSIGM(6),DEPT(3,3),EPM(6),EPB(6),CB(6,6)
      DATA II/11,22,33,12,13,23/, DLTA/1.0,3*0.0,1.0,3*0.0,1.0/
      ALFUN(CV,RT,SINV)=2.0*RT*CV/(1.0+RT-(1.0-RT)*SINV)
      SMALL=0.0001*PROP(8)
      DO 40 I=1,6
      SIGB(I)=SIGBM(I)
      DSIG(I)=DSIGM(I)
      EPB(I)=EPM(I)
   40 DEP(I)=DEPM(I)
      IF(ITNO.GT.1) GO TO 100
      IF(INC .GT.1) GO TO 50
```

```
C
C     INITIALIZE HISTORY
C
      STOR(1)=PROP(21)
      STOR(2)=STOR(1)
      STOR(3)=0.5*(SIGB(1)+SIGB(2))
      STOR(4)=0.01*PROP(8)
      STOR(5)=0.0
      GO TO 100
C
C     UPDATE HISTORY
C
   50 STOR(1)=STOR(2)
      STOR(3)=STOR(3)+STOR(4)
      STOR(5)=STOR(5)+STOR(6)
C
C     CONVERT FROM PLANE STRAIN TO 3-D
C
  100 IF(IDIM.EQ.3) GO TO 200
      SIGB(4)=SIGB(3)
      SIGB(3)=STOR(3)
      DSIG(4)=DSIG(3)
      DSIG(3)=STOR(4)
      DEP(4)=DEP(3)
      DEP(3)=0.0
      EPB(4)=EPB(3)
      EPB(3)=0.0
      DO 110 I=5,6
      SIGB(I)=0.0
      DSIG(I)=0.0
      EPB(I)=0.0
  110 DEP(I)=0.0
C
C     DETERMINE 3-D INCREMENTAL PROPERTIES
C
  200 GAM=PROP(6)
C
C     CALCULATE EFFECTIVE STRESS INVARIANTS AND DISTORTIONAL STRESS
C     AND CHANGE MATRIX COMPONENTS TO TENSOR COMPONENTS.
C
      XIB=0.0
      XIF=0.0
      DDIL=0.0
      DILB=0.0
      DO 205 I=1,3
      DDIL=DDIL+DEP(I)
      DILB=DILB+EPB(I)
      XIB=XIB+SIGB(I)
  205 XIF=XIF+SIGB(I)+DSIG(I)
      VOIDB=1.0+PROP(20)
      VOIDF=VOIDB
      IF(LARGE.EQ.0) GO TO 210
      VOIDB=VOIDB*EXP(-DILB)
      VOIDF=VOIDF*EXP(-DILB-DDIL)
```

```
  210 DO 215 N=1,6
      I=II(N)/10
      J=MOD(II(N),10)
      SB(I,J)=SIGB(N)-XIB*DLTA(I,J)/3.0
      SB(J,I)=SB(I,J)
      DEPT(I,J)=DEP(N)*(1.0+DLTA(I,J))*0.5
      DEPT(J,I)=DEPT(I,J)
      SF(I,J)=SIGB(N)+DSIG(N)-DLTA(I,J)*XIF/3.0
  215 SF(J,I)=SF(I,J)
      GAMP=0.0
      IF(KIND .EQ. 0)GO TO 217
      GAMP=GAM
      UB=STOR(5)
      DLTAU=GAM*DDIL
  217 XIB=XIB-UB*3.0
      XIF=XIF-(UB+DLTAU)*3.0
      STOR(6)=DLTAU
      SRTJB=0.0
      SRTJF=0.0
      DO 220 I=1,3
      DO 220 J=1,3
      SRTJB=SRTJB+SB(I,J)*SB(I,J)
  220 SRTJF=SRTJF+SF(I,J)*SF(I,J)
      SRTJB=SQRT(0.5*SRTJB)
      IF(SRTJB*1000. .LT. XIB)SRTJB=0.0
      SRTJF=SQRT(0.5*SRTJF)
      IF(SRTJF*1000. .LT. XIF)SRTJF=0.0
      SCUB=0.0
      SCUF=0.0
      DO 225 I=1,3
      DO 225 J=1,3
      DO 225 K=1,3
      SCUB=SCUB+SB(I,J)*SB(J,K)*SB(K,I)
  225 SCUF=SCUF+SF(I,J)*SF(J,K)*SF(K,I)
      SCUB=SCUB/3.0
      SCUF=SCUF/3.0
      SN3AB=0.0
      IF(SRTJB.GT.SMALL) SN3AB=1.5*SQRT(3.0)*SCUB/SRTJB**3
      IF(SN3AB.GT. 1.0) SN3AB= 1.0
      IF(SN3AB.LT.-1.0) SN3AB=-1.0
      SN3AF=0.0
      IF(SRTJF.GT.SMALL) SN3AF=1.5*SQRT(3.0)*SCUF/SRTJF**3
      IF(SN3AF.GT. 1.0) SN3AF= 1.0
      IF(SN3AF.LT.-1.0) SN3AF=-1.0
      CS3AB=SQRT(1.0-SN3AB**2)
      CS3AF=SQRT(1.0-SN3AF**2)
C
C     AVOID ZERO MEAN PRESSURE
C
      IF(ABS(XIB).GT.SMALL) GO TO 227
      DU=XIB
      XIB=SMALL
      IF(DU.LT.0.0) XIB=-SMALL
  227 IF(ABS(XIF).GT.SMALL) GO TO 230
```

```
      DU=XIF
      XIF=SMALL
      IF(DU.LT.0.0) XIF=-SMALL
  230 CONTINUE
C
C     CALCULATE ELASTIC PROPERTIES
C
      DU1=VOIDB/3.0/PROP(2)
      DU2=1.5*(1.0-2.0*PROP(5))/(1.0+PROP(5))
      DU=XIB
      IF(DU.LT.PROP(7)) DU=PROP(7)
      BB=DU1*DU
      GB=DU2*BB
      IF(PROP(5).GT.0.5) GB=PROP(5)
      DU1=VOIDF/3.0/PROP(2)
      DU=XIF
      IF(DU.LT.PROP(7)) DU=PROP(7)
      BF=DU1*DU
      GF=DU2*BF
      IF(PROP(5).GT.0.5) GF=GB
      DO 235 M=1,6
      I=II(M)/10
      J=MOD(II(M),10)
      DO 235 N=M,6
      K=II(N)/10
      L=MOD(II(N),10)
      DU1=DLTA(K,I)*DLTA(L,J)+DLTA(K,J)*DLTA(I,L)
      C(M,N)=GF*DU1+(BF+GAMP-2.0*GF/3.0)*DLTA(I,J)*DLTA(K,L)
      CB(M,N)=GB*DU1+(BB+GAMP-2.0*GB/3.0)*DLTA(I,J)*DLTA(K,L)
      CB(N,M)=CB(M,N)
  235 C(N,M)=C(M,N)
C
C     CALCULATE SIZE OF BOUNDING SURFACE
C
      XIOB=STOR(1)
      XIOF=STOR(2)
      XIL=PROP(7)
      DU10=1.0/(PROP(1)-PROP(2))
      IF(XIOB.GE.XIL.AND.XIOF.GE.XIL) GO TO 240
      XIOBS=XIOB
      IF(XIOB.LT.XIL) XIOBS=XIL
      XIOFS=XIOF
      IF(XIOF.LT.XIL) XIOFS=XIL
      XIOF=XIOB+DU10*0.5*((XIOFS*VOIDF+XIOBS*VOIDB)*DDIL-
     1     (XIOBS*VOIDB/BB+XIOFS*VOIDF/BF)*(XIF-XIB)/3.0)
      GO TO 245
  240 XIOF=XIOB*EXP(DU10*0.5*((VOIDB+VOIDF)*DDIL-
     1     (VOIDB/BB+VOIDF/BF)*(XIF-XIB)/3.0))
  245 STOR(2)=XIOF
      IF(INC+ITNO.EQ.2) GO TO 410
C
C     CALCULATE BOUNDING SURFACE PROPERTIES
C
      CALL BOUND(PROP,SRTJB,SN3AB,XSB,XIOB,XIB,GAMB,DFIB,
```

```
     1           DFJB,XKSB,DFALB,DFJJB,BSB,VOIDB)
      CALL BOUND(PROP,SRTJF,SN3AF,XSF,XIOF,XIF,GAMF,DFIF,
     1           DFJF,XKSF,DFALF,DFJJF,BSF,VOIDF)
      DB=BSB-1.0
      IF(DB .LT. 0.0) DB=0.0
      DF=BSF-1.0
      IF(DF.LT.0.0) DF=0.0
C
C     CALCULATE PLASTIC MODULUS
C     CHECK FOR ELASTIC ZONE AND UNLOADING
C
      XMS=ALFUN(PROP(17),PROP(19),SN3AB)
      DU7=0.0001**(1.0/XMS)
      LB=0
      DDD=1.0+DB*(1.0-PROP(15))
      IF(DDD.LE.0.0) GO TO 352
      LB=1
      H=ALFUN(PROP(16),PROP(18),SN3AB)
      DU=ABS(XSB)
      IF(DU.LT.DU7) DU=DU7
      DU8=9.0*DFIB**2+DFJB**2/3.0
      DU9=XIOB
      IF(XIOB.LT.XIL) DU9=XIL
      XKB=XKSB+H*DB/DDD*(1.0+1.0/DU**XMS)*DU8*DU9*DU10*VOIDB
      DU1=3.0*BB*DFIB
      DU2=GB*DFJJB
      DU2P=SQRT(3.0)*GB*DFALB
      DU3=XKB+9.0*BB*DFIB**2+GB*DFJB**2+GB*(DFALB*CS3AB)**2
      SUM=0.0
      T1=0.0
      IF(SRTJB**2 .EQ.0.0) GO TO 350
      DO 340 I=1,3
      DO 340 J=1,3
      DU=0.0
      DO 330 K=1,3
  330 DU=DU+SB(I,K)*SB(K,J)
      T1=T1+(DU-1.5*SCUB*SB(I,J)/SRTJB**2)*DEPT(I,J)/SRTJB**2
  340 SUM=SUM+SB(I,J)*DEPT(I,J)
      T1=T1-2.0*DDIL/3.0
  350 DU=(DU1*DDIL+DU2*SUM+DU2P*T1)/DU3
      IF(DU.LT.0.0) LB=0
  352 LF=0
      DDD=1.0+DF*(1.0-PROP(15))
      IF(DDD.LE.0.0) GO TO 358
      LF=1
      H=ALFUN(PROP(16),PROP(18),SN3AF)
      DU=ABS(XSF)
      IF(DU.LT.DU7) DU=DU7
      DU8=9.0*DFIF**2+DFJF**2/3.0
      XMS=ALFUN(PROP(17),PROP(19),SN3AF)
      DU9=XIOF
      IF(XIOF.LT.XIL) DU9=XIL
      XKF=XKSF+H*DF/DDD*(1.0+1.0/DU**XMS)*DU8*DU9*DU10*VOIDF
      DU4=3.0*BF*DFIF
```

```
      DU5=GF*DFJJF
      DU6=XKF+9.0*BF*DFIF**2+GF*DFJF**2+GF*(DFALF*CS3AF)**2
      DU5P=GF*DFALF*SQRT(3.0)
      SUM=0.0
      T1=0.0
      IF(SRTJF**2 .EQ.0.0) GO TO 357
      DO 356 I=1,3
      DO 356 J=1,3
      DU=0.0
      DO 355 K=1,3
  355 DU=DU+SF(I,K)*SF(K,J)
      T1=T1+(DU-1.5*SCUF*SF(I,J)/SRTJF**2)*DEPT(I,J)/SRTJF**2
  356 SUM=SUM+SF(I,J)*DEPT(I,J)
      T1=T1-2.0*DDIL/3.0
  357 DU=(DU4*DDIL+DU5*SUM+DU5P*T1)/DU6
      IF(DU.LT.0.0) LF=0
C
C     CALCULATE PLASTIC PORTION OF INCREMENTAL PROPERTIES
C
  358 IF(LF+LB.EQ.0) GO TO 410
      DO 400 M=1,6
      I=II(M)/10
      J=MOD(II(M),10)
      DO 400 N=M,6
      K=II(N)/10
      L=MOD(II(N),10)
      DU=0.0
      IF(LB.EQ.0) GO TO 373
      T2=0.0
      T1=0.0
      IF(SRTJB**4 .EQ.0.0) GO TO 370
      DO 360 LL=1,3
      T2=T2+SB(K,LL)*SB(LL,L)
  360 T1=T1+SB(I,LL)*SB(LL,J)
      T1=DU2P*(T1/SRTJB**2-1.5*SCUB*SB(I,J)/SRTJB**4-2.0*DLTA(I,J)/3.0)
      T2=DU2P*(T2/SRTJB**2-1.5*SCUB*SB(K,L)/SRTJB**4-2.0*DLTA(K,L)/3.0)
  370 DUB=-(DU1*DLTA(I,J)+DU2*SB(I,J)+T1)*(DU1*DLTA(K,L)+
     1    DU2*SB(K,L)+T2)/DU3
      IF(LF.EQ.0) GO TO 396
  373 T2=0.0
      T1=0.0
      IF(SRTJF**4 .EQ.0.0) GO TO 390
      DO 380 LL=1,3
      T2=T2+SF(K,LL)*SF(LL,L)
  380 T1=T1+SF(I,LL)*SF(LL,J)
      T1=DU5P*(T1/SRTJF**2-1.5*SCUF*SF(I,J)/SRTJF**4-2.0*DLTA(I,J)/3.0)
      T2=DU5P*(T2/SRTJF**2-1.5*SCUF*SF(K,L)/SRTJF**4-2.0*DLTA(K,L)/3.0)
  390 DU=-(DU4*DLTA(I,J)+DU5*SF(I,J)+T1)*(DU4*DLTA(K,L)+
     1      DU5*SF(K,L)+T2)/DU6
  396 C(M,N)=DU+C(M,N)
      CB(M,N)=DUB+CB(M,N)
      CB(N,M)=CB(M,N)
  400 C(N,M)=C(M,N)
  410 CONTINUE
```

```
      IF(IDIM.EQ.3) RETURN
C
C     CONVERT 3-D PROPERTIES TO PLANE STRAIN
C
      DU=0.0
      DO 420 I=1,4
      DU=C(3,I)*DEP(I)+DU
      C(3,I)=C(4,I)
  420 C(4,I)=0.0
      DO 430 I=1,3
      C(I,3)=C(I,4)
  430 C(I,4)=0.0
      STOR(4)=DU
      RETURN
      END
C*****************************************************************
      SUBROUTINE BOUND(PROP,SRTJ,SN3A,X,XIO,XI,GAM,DFI,DFJ,
     1                 XKS,DFAL,DFJJ,BS,VOID)
C
C     SUBROUTINE TO EVALUATE RELATIONSHIP OF STRESS STATE
C     TO BOUNDING SURFACE
C
      DIMENSION PROP(21),FSS(3)
      ALFUN(CV,RT,SINV)=2.0*RT*CV/(1.0+RT-(1.0-RT)*SINV)
      DFUN(FUN,RT,FUNC)=FUN**2*(1.0-RT)/(2.0*RT*FUNC)
      XN=ALFUN(PROP(3),PROP(4),SN3A)
      DNAL=DFUN(XN,PROP(4),PROP(3))
      R=ALFUN(PROP(9),PROP(12),SN3A)
      DRAL=DFUN(R,PROP(12),PROP(9))
      A=ALFUN(PROP(10),PROP(13),SN3A)
      DAAL=DFUN(A,PROP(13),PROP(10))
      YS=R*A/XN
      CC=PROP(14)
C
C     SHIFT PROJECTION POINT
C
      D1=XI-XIO*CC
      IF(ABS(D1).LT.0.001) D1=0.001
      D2=CC-1.0/R
      D3=D1*D2
      D5=CC*(CC-2.0/R)
      Q =SRTJ/D1
      QC=XN/(1.0-R*CC)
      QO=1.0E+20
      IF(CC.NE.0.0) QO=XN*(SQRT(1.0+YS*YS)-(1.0+YS))/R/CC
      IF(SRTJ.NE.0.0) GO TO 3
      IF(D1.GT.0.0) GO TO 10
      GO TO 30
    3 IF(CC.LT.1.0/R) GO TO 5
      IF(Q .GE.0.0) GO TO 10
      IF(Q .LE. QC) GO TO 10
      IF(Q .GE. QO) GO TO 30
      GO TO 20
    5 IF(Q .GE. QC) GO TO 20
```

```
      IF(Q .GE.0.0) GO TO 10
      IF(Q .LE. QO) GO TO 20
      GO TO 30
C
C     PROJECTION ON ELLIPSE 1
C
   10 D4=D1*D1+((R-1.0)*SRTJ/XN)**2
      BS=XIO*(-D3+SQRT(D3*D3-D4*(D5+(2.0-R)/R)))/D4
      LOST=1
      GO TO 100
C
C     PROJECTION ON HYPERBOLA
C
   20 D6=SRTJ*(1.0/R+A/XN)/XN
      D7=D3+D6
      D8=D1*D1-(SRTJ/XN)**2
      BS=-0.5*XIO*(D5-2.0*A/R/XN)/D7
      IF(D8.EQ.0.0) GO TO 100
      BS=XIO*(-D7+SQRT(D7*D7-D8*(D5-2.0*A/R/XN)))/D8
      LOST=2
      GO TO 100
C
C     PROJECTION ON ELLIPSE 2
C
   30 T=PROP(11)
      FOP=XN/SQRT(1.0+YS**2)
      XJO=A*(1.0+YS-SQRT(1.0+YS**2))/YS
      BT=T*(XJO-T*FOP)/(XJO-2.0*T*FOP)
      RO=(BT-T)/FOP/XJO
      PSI=1.0/(R*(BT-T))
      D9=T-BT+CC
      D10=D1*D9
      D11=D1*D1+RO*SRTJ*SRTJ
      BS=XIO*(-D10+SQRT(D10*D10-D11*(D9*D9-BT*BT)))/D11
      LOST=3
  100 XIBAR=BS*(XI-XIO*CC)+XIO*CC
      IF(XIBAR.EQ.0.0) XIBAR=1.0E-20
      TH=BS*SRTJ/XIBAR
      X=TH/XN
      DU=XIO
      IF(XIO.LT.PROP(7)) DU=PROP(7)
      DUS=12.0*VOID/(PROP(1)-PROP(2))*XIO**2*DU
      GO TO (110,200,300),LOST
C
C     NORMAL CONSOLIDATION ZONE
C
  110 PSI=YS/(R-1.0)**2
      DU=R*(1.0+X*X+R*(R-2.0)*X*X)
      GAM=(1.0+(R-1.0)*SQRT(1.0+R*(R-2.0)*X*X))/DU
      DFI=2.0*XIO*(GAM-1.0/R)*PSI
      DFJJ=2.0*XIO*GAM*((R-1.0)/XN)**2*PSI*BS/XIBAR
      DFJ=DFJJ*SRTJ
      XKS=DUS*(GAM-1.0/R)*(GAM+R-2.0)*PSI*PSI/R
      DFAL=PSI*6.0*(R-1.0)*TH*GAM*XIO*(((R-1.0)/(R**2*
```

```
     1      (2.0/R-GAM-1.0))+1.0)*DRAL-(R-1.0)*DNAL/XN)/XN**2
       RETURN
C
C      OVERCONSOLIDATED ZONE
C
  200 DU=1.0-X*(1.0+YS)
       GAM=-(DU+SQRT((X-YS-1.0)**2+(X*X-1.0)*YS*YS))/(R*(X*X-1.0))
       DFI=2.0*XIO*(GAM-1.0/R)
       DFJ=2.0*XIO*((1.0+YS)/R-X*GAM)/XN
       XKS=DUS*(GAM-1.0/R)*(DU*GAM+2.0*A/XN)/R
       DFJJ=DFJ/SRTJ
       DFAL=6.0*XIO*(DNAL*(TH*GAM/XN-1.0/R+A/(R*TH*GAM)-2.0*A/XN)/
     1      XN**2+DRAL*(1.0/TH-1.0/XN+A/(XN*TH*GAM))/R**2+DAAL*(
     2      1.0/XN-1.0/(TH*GAM*R))/XN)
       RETURN
C
C      TENSION ZONE
C
  300 GAM=(-T+BT-SQRT(BT*BT-RO*TH*TH*T*(T-2.0*BT)))/(1.0+RO*TH*TH)
       DFI=2.0*PSI*XIO*(GAM+T-BT)
       DFJJ=2.0*PSI*XIO*GAM*RO*BS/XIBAR
       DFJ=DFJJ*SRTJ
       XKS=DUS*PSI*PSI*(GAM+T-BT)*(GAM*(BT-T)+T*(2.0*BT-T))
       DYSAL=YS*(DRAL/R+DAAL/A-DNAL/XN)
       DFOPAL=FOP*(DNAL/XN-YS*DYSAL/(1.0+YS*YS))
       DJOAL=XJO*(DAAL/A-DYSAL/YS)+A*(1.0/YS-FOP/XN)*DYSAL
       DBTAL=((T-BT)*DJOAL-(T-2.0*BT)*T*DFOPAL)/(XJO-2.0*T*FOP)
       DROAL=DBTAL/FOP/XJO-RO*(DFOPAL/FOP+DJOAL/XJO)
       DFAL=3.0*PSI*XIO*TH*GAM*(DROAL+2.0*RO*DBTAL/(T+GAM-2.0*BT))
       RETURN
       END
C*****************************************************************
       SUBROUTINE ACCEL(X2,X1,X,C,XL)
C
C      THIS SUBROUTINE CALCULATES THE AITKENS CONVERGENCE FACTOR
C
       C=1.0
       DU=-X2+2.0*X1-X
       IF(DU .EQ. 0.0)RETURN
       C=(X1-X2)/DU
       IF(C .GT. XL)C=XL
       IF(C .LT. 1.0/XL)C=1.0/XL
       RETURN
       END
```